RMC70 Motion Controller

And

RMC70Tools Software

User Manual

Version 1.10.0 May 7, 2004

Condensed Contents

1. Introducing the RMC70 Series 1
Overview of the RMC70 capabilities and applications.

2. Starting Up the RMC70 5
A Step-by-step guide to quickly get up and running.

3. Control Features 19
Control Types and more…

4. Programming 37
A guide to programming the RMC70.

5. Using RMC70Tools 57
Detailed information on how to use the RMC70 motion control software.

6. Modules 79
CPU, Axis and Expansion modules.

7. Registers 107
Detailed information on all RMC70 registers.

8. Communication Types 113
Serial, PROFIUBUS, and more…

9. Troubleshooting 131
Hints and assistance. Detailed information on error codes.

10. Command Reference 181
Detailed information on how to use each RMC70 command.

11. RMC70 Specifications 215
Hardware specifications of the RMC70.

12. Index 239

Table Of Contents

Introducing the RMC70 Series 1

RMC70 Series Motion Controllers 1
RMC70 Physical Components 1
RMC70 Axis Types 2

Starting Up the RMC70 5

RMC70 Start-up Procedure 5
Scaling 11
Tuning 11
Tuning Overview 12
Tuning a Position Axis 13

Control Features 19

RMC70 Control Features 19
Halts 19
Halts Overview 19
External Halt 21
Closed Loop Halt 21
Open Loop Halt 22
Open Loop with Disable Output Halt 22
Control Modes 23
Control Modes Overview 23
Closed Loop Control 24
Open Loop Control 24
Pressure Limit 25
Position-Pressure Control 26
PFID Control 26
Plots 28
Using Plots 29
Advanced Plot Capabilities 29
Configuring Plots 30
Triggering Plots 32
Selecting Data to Plot 33
Assigning Plots to Axes 34
Using Plots with a Host Controller 35

Programming 37

User Programs 37
User Programs Overview 37
Creating User Programs 38
Verifying User Programs 40
Running User Programs 40
Labeling Steps 41
Expressions 42
Expressions Overview 42
Assignment Expressions 42
Logical Expressions 43
Functions 44
Operators 45
Troubleshooting Expressions 46
Link Types 47

Link Types Overview ... 47
Link Type: End ... 48
Link Type: Immediate ... 48
Link Type: Delay ... 48
Link Type: Wait For ... 49
Link Type: Jump ... 49
Link Type: Delay Jump ... 50
Link Type: Conditional Jump ... 50
Programming Overview ... 51
Tasks ... 52
Variables ... 53
PreScan Table ... 55
Data Types ... 56

Using RMC70Tools ... 57

Using RMC70Tools ... 57
RMC70Tools IDE ... 57
RMC70Tools Menu Bar ... 58
RMC70Tools Project Pane ... 59
Axis Tools Pane ... 60
Event Log Monitor ... 61
Uploading and Downloading Axes ... 62
Command Tool ... 62
Output Pane ... 63
Axes Status Registers Pane ... 63
Hiding and Showing Columns ... 64
Step Editor ... 64
Program Monitor ... 64
Shortcut Keys ... 65
Programming ... 66
Go Online, GO Offline ... 68
Editors ... 68
Axes Parameters Pane ... 68
Indirect Data Map Editor ... 70
Custom Data Map Editor ... 71
Using Editors ... 71
Variable Table Editor ... 73
Plot Manager ... 75
Plot Manager: Viewing Plots ... 75
Plot Manager Overview ... 76
Plot Manager Elements ... 77

Modules ... 79

CPU Modules ... 79
RMC75S ... 79
RMC75S CPU Module ... 79
RMC75S Wiring ... 79
RMC75P ... 79
RMC75P CPU Module ... 79
RMC75P Wiring ... 79
CPU Modules Overview ... 80
CPU Module LEDs ... 80
Axis Modules ... 82
Axis Modules Overview ... 82
Control Output ... 82
Enable Output ... 83
Fault Input ... 84

AA 85
AA Module 85
AA Axis Module Specifications 85
AA Wiring 86
AA LEDs 88
Analog Scaling for the AA Module 89
MA 90
MA Axis Module 90
MAx Supported Transducer Options 90
MA Axis Module Specifications 91
MAx Wiring 92
MA LEDs 96
MDT Scaling 96
MDT Transducer Fundamentals 98
SSI Scaling 99
SSI Transducer Fundamentals 100
Expansion Modules 101
Expansion Modules Overview 101
AP2 102
AP2 Expansion Module 102
AP2 Wiring 103
AP2 LEDs 104
Analog Scaling for the AP Module 104

Troubleshooting 107

Troubleshooting Overview 107
Error Codes 107
Technical Support 111

Registers 113

RMC70 Register Address Format 113
RMC70 Register Map 114
Status Registers 127
Axis Status Registers Overview 127
Common 127
Error Bits Register 127
Last Error Number Register 129
Read Response 130
Status Bits Register 131
Input 133
Actual Acceleration 133
Actual Differential Force 134
Actual Differential Force Rate 134
Actual Force A, Actual Force B 135
Actual Position 135
Actual Pressure/Force 135
Actual Pressure/Force Rate 136
Actual Velocity 136
Counts 136
Volts 137
Counts A, Counts B 137
Raw Counts 137
Raw Counts A, Raw Counts B 138
Output 138
Control Output 138
Primary Control 138
Acceleration Feed Forward Term 138
Differential Output Term 139
Integral Output Term 139
PFID Output 139

Position Error ... 139
Proportional Output Term ... 140
Velocity Error ... 140
Velocity Feed Forward Term ... 140
Jerk Feed Forward Term ... 141
Primary Target ... 141
Command Position ... 141
Command Velocity ... 141
Target Acceleration ... 142
Target Position ... 142
Target Velocity ... 142
Secondary Control ... 143
Pressure/Force Differential Term ... 143
Pressure/Force Error ... 143
Pressure/Force Feed Forward Term ... 143
Pressure/Force Proportional Term ... 143
Pressure/Force Rate Feed Forward Term ... 144
Secondary Target ... 144
Command Pressure/Force ... 144
Target Pressure/Force ... 144
Parameter Registers ... 145
Axis Parameter Registers Overview ... 145
Feedback ... 145
Input Filters ... 145
Actual Acceleration Filter ... 145
Actual Position Filter ... 145
Actual Pressure/Force Filter ... 146
Actual Pressure/Force Rate Filter ... 146
Actual Velocity Filter ... 146
Pressure & Force ... 147
Force A Offset, Force B Offset ... 147
Force A Scale, Force B Scale ... 147
Pressure/Force Offset ... 148
Pressure/Force Scale ... 148
Transducer Specific ... 148
Feedback Type ... 148
MDT Type ... 149
Absolute/Incremental ... 152
Noise Error Rate ... 152
Position Offset ... 153
Position Scale ... 153
Linear/Rotary ... 154
Stop Threshold ... 154
Velocity Deadband ... 154
Velocity Offset ... 155
Halting ... 155
Auto Stop Configuration ... 155
Halt Group Number ... 157
Closed Loop Halt Deceleration ... 157
Open Loop Halt Ramp ... 157
Output ... 158
Deadband Tolerance ... 158
Directional Gain Ratio ... 159
Invert Output Polarity ... 160
Output Bias ... 160
Output Deadband ... 160
Output Filter ... 162
Output Limit ... 162
Output Scale ... 163
Fault Input Polarity ... 163
Enable Output Behavior ... 163

Primary Control ... 164
In Position Tolerance ... 164
Position Error Tolerance ... 164
At Velocity Tolerance ... 165
Jerk Feed Forward ... 165
Velocity Error Tolerance ... 166
Integrator Mode ... 166
Position Proportional Gain ... 167
Position Integral Gain ... 167
Position Differential Gain ... 169
Velocity Feed Forward ... 170
Acceleration Feed Forward ... 171
Simulation ... 171
Simulate Mode ... 171
System Gain ... 173
Natural Frequency ... 173
Damping Factor ... 174
Secondary Control ... 174
At Pressure/Force Tolerance ... 174
Pressure/Force Error Tolerance ... 174
Pressure/Force Proportional Gain ... 175
Pressure/Force Integral Gain ... 175
Pressure/Force Differential Gain ... 176
Pressure/Force Feed Forward ... 176
Pressure/Force Rate Feed Forward ... 177
Pressure/Force Orientation ... 177
Target ... 178
Positive Travel Limit ... 178
Negative Travel Limit ... 179
Positive Pressure/Force Limit ... 179
Negative Pressure/Force Limit ... 179

Communication Types ... 181

RMC70 Communications Overview ... 181
Indirect Data Map ... 181
RS-232 Monitor Port ... 182
PROFIBUS ... 184
PROFIBUS-DP Communications Overview ... 184
PROFIBUS Configuration ... 185
PROFIBUS Mode 8-1 ... 187
PROFIBUS Mode 16-1 ... 192
PROFIBUS Mode 8-7 ... 193
Configuring a PROFIBUS-DP Network with COM PROFIBUS ... 194
Configuring a PROFIBUS-DP Network with SST Profibus Configuration ... 196
Configuring a PROFIBUS-DP Network with SyCon ... 197
Serial (RS-232/485) ... 198
Serial Communications Overview ... 198
Configuration and Wiring ... 200
Line Drivers: RS-232/485 ... 200
Configuring the RMC7xS Serial Communications ... 201
Serial Network Topologies ... 202
RS-232 Wiring for the RMC70 ... 204
RS-485 Wiring for the RMC70 ... 205
RS-485 Termination and Biasing ... 207
RMC70 Serial Protocols ... 209
DF1 Protocol (Full-and Half-Duplex) ... 209
Modbus/RTU Protocol ... 213

Command Reference ... 215

RMC70 Commands Overview **215**
RMC70 Commands **215**
General Commands **217**
RMC70 Command: No-op (0) 217
RMC70 Command: Clear Faults (4) 217
RMC70 Command: Initialize Axes (7) 217
RMC70 Command: Fault Controller (8) 218
RUN Mode (98) 219
PROGRAM Mode (99) 219
Motion Commands **220**
Motion Commands 220
Open Loop 221
RMC70 Command: Direct Drive (9) 221
RMC70 Command: Open Loop Rate (10) 221
RMC70 Command: Open Loop Absolute (11) 222
RMC70 Command: Open Loop Relative (12) 223
Point-to-Point 224
RMC70 Command: Move Absolute (20) 224
RMC70 Command: Move Relative (21) 224
RMC70 Command: Quick Move Absolute (15) 225
RMC70 Command: Quick Move Relative (16) 226
RMC70 Command: Time Move Absolute (23) 226
RMC70 Command: Time Move Relative (24) 227
Specialty 227
RMC70 Command: Speed at Position (36) 227
Stops 228
RMC70 Command: Closed Loop Halt (1) 228
RMC70 Command: Open Loop Halt (2) 228
RMC70 Command: Disable Output (3) 229
RMC70 Command: Hold Current Position (5) 229
RMC70 Command: Stop (6) 229
System **230**
RMC70 Command: Update Flash (110) 230
Pressure Control **230**
RMC70 Command: Enable/Disable Pressure Limit (40) 230
RMC70 Command: Ramp Pressure(41) 231
Set Parameters **231**
RMC70 Command: Set Integrator Mode (71) 232
RMC70 Command: Set Enable Output (67) 232
RMC70 Command: Set Actual Position (49) 232
RMC70 Command: Set Target Position (48) 233
RMC70 Command: Read Register (111) 233
RMC70 Command: Write Register (112) 233
Programming **234**
RMC70 Command: Stop Task (91) 234
RMC70 Command: Start Task (90) 234
Plots **235**
RMC70 Command: Enable/Disable Plot Trigger (104) 235
RMC70 Command: Rearm Plot (103) 236
RMC70 Command: Start Plot (100) 236
RMC70 Command: Stop Plot (101) 236
RMC70 Command: Trigger Plot (102) 237
Step Editor Commands **237**
RMC70 Command: Expression (113) 237

RMC70 Specifications **239**

RMC70 Specifications **239**
Wiring Guidelines **239**

RMC70 Part Numbering **240**

Index **243**

RMC70Tools On-line Help

The RMC70Tools on-line help provides the most complete and up-to-date information available for the RMC70 and RMC70Tools. To ensure you have the latest help file, download the latest version of RMC70Tools from Delta's website www.deltamotion.com.

Troubleshooting

See the Troubleshooting topic.

Customer Support

Delta support is ready to assist you. Call should you have questions or need help:

Telephone:	360-254-8688
Fax:	360-254-5435
24-hour paging:	360-699-7784
E-mail:	support@deltamotion.com
Website:	http://www.deltamotion.com

Paper Copy of On-line Help

A manual containing the same information as the online help is available as a Portable Document Format (PDF) file from Delta's website www.deltamotion.com. If you need a hard-copy manual, contact Delta by telephone at 360-254-8688 or email at sales@deltamotion.com.

Disclaimer

Although great effort has been taken to ensure the accuracy of the information in this documentation, it is intended to be used only as a guide. Knowledge of motion control, hydraulic servos, electric servos, transducers, and safety rules is required. Delta Computer Systems, Inc. does not accept responsibility for problems resulting from errors or omissions in this documentation. The information in this documentation is subject to change without notice.

Neither Delta Computer Systems, Inc. nor anyone else involved in the creation, production, or delivery of this product shall be liable for any direct, indirect, consequential injuries and or damages arising out of the use, the results of use, or the inability to use this product.

All brand names and trademarks referenced in this manual are the property of their respective owners.

RMC70 Warranty

GENERAL

These general terms and conditions of sale of Delta Computer Systems, Inc (Delta), along with any written Delta quotation, exclusively will govern the sale or licensing by Delta of all goods and services (including hardware, firmware, and software products, training, support, parts and repair services) furnished hereunder. No additions or modifications will be binding on Delta unless agreed to in writing by an authorized representative of Delta.

WARRANTY

Hardware products will conform to Deltas material specifications and be free of defects in material and workmanship under normal and proper use for a period of one (1) year from date of shipment to Customer by Delta or its authorized distributor. Repaired or replacement products are similarly warranted for six (6) months or the remainder of the original warranty term, whichever is longer.

Non-warranty repaired or replacement products (which may be new or reconditioned) are warranted to be free of defects in materials and workmanship for six (6) months from date of shipment from Delta.

Standard firmware and software will perform in accordance with Deltas published specifications when used with Delta-specified hardware for a period of one (1) year from date of shipment to Customer by Delta or its authorized distributor. Delta makes no representation or warranty, express or implied, that the operation of the firmware or software will be uninterrupted or error-free, or will meet or satisfy the Customers intended use or requirements. Corrections are warranted for a period of six (6) months or the remainder of the original warranty term, whichever is longer.

Deltas sole liability under this warranty shall be, at Deltas discretion, to repair or replace product found defective or to issue Customer credit for the purchase price of defective product. Customer may obtain service under this warrant if: a) within the warranty period Customer notifies Delta of the defective product and obtains a Return Material Authorization (RMA) from Delta; b) Customer returns product to Delta, postage prepaid, in compliance with the RMA instructions, and c) Delta, upon inspection, confirms the existence of the defect, and determines that the product has not been subject to misuse, neglect, accident, or improper installation, operation, or application, or has been repaired or altered by others. Any warrantee service (consisting of time, travel and expenses relating to such services) performed other than at Delta, will be at Customers expense.

THIS WARRANTY IS IN LIEU OF ANY OTHER WARRANTY, EXPRESS OR IMPLIED. DELTA DISCLAIMS ANY IMPLIED WARRANTY OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. DELTAS RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY.

LIMITATION OF LIABILITY

IN NO EVENT SHALL DELTA OR ITS DISTRIBUTORS OR VENDORS BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR CONNECTED WITH CUSTOMERS PURCHASE OR USE OF ANY PRODUCT, EVEN IF DELTA OR THE DISTRIBUTOR OR VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

RMC Return for Repair

Should an RMC require repair, refer to the Technical Support topic for return details.

1. Introducing the RMC70 Series

1.1. RMC70 Series Motion Controllers

The one- and two-axis RMC70 series motion controllers brings the benefits of high-performance, easy-to-use motion control to a wide range of industrial applications.

1.2. RMC70 Physical Components

This topic describes the physical components of the RMC70 on a basic informational level. For mechanical and electrical specifications, see the RMC70 Specifications. For detailed function of each module, see the respective links.

Controller Modules

An RMC70 motion controller consists of the following physical components:

- **CPU Module**

  This module includes the main motion control processing unit. It also incorporates the primary communication channel such as serial, DeviceNet, Ethernet, etc. and a serial port for communications to the RMC70Tools software.

- **Axis Modules**

  This module interfaces to a transducer and the axis drive. It is attached to the Control Module and is sold as one unit with the Control Module. Axis Modules are transducer specific.

> **Note:**
> The Control Module and Axis Module are one unit (called a Base Module) and make up a complete motion controller. They are referred to as separate units solely for the purpose of explaining their functionality.

- **Optional Expansion Modules**

  Up to four optional expansion modules can be added to enhance the capabilities of the RMC70. These modules are field-changeable. Currently, an analog input expansion module is available.

Example Module

The module below consists of a RMC75S-MA2 base module and an AP2 expansion module.

CPU Module **Axis Module** **Expansion Module**

1.3. RMC70 Axis Types

The RMC70 is a 1- or 2-axis motion controller with up to 3 additional reference axes (half-axes). The axis types listed in this topic are or will be available on the RMC70. The required inputs and outputs for each axis type constitute 1 axis. For example, a Position-Force application may require 3 inputs and 1 output, but is still considered one axis.

Maximum Number of Axes

The maximum number of control axes on the RMC70 is 2. The maximum number of total axes (control axes plus reference axes) is 4. A reference axis is also commonly known as a half-axis.

Control Axes

Control axes have a Control Output and an input for transducer feedback. A control axis is capable of controlling a system because it has a Control Output. All axes on the RMC70 Axis Modules are control axes.

Control Axis Types	Required Inputs	Required Outputs	Transducer Types	Possible RMC70 Modules
Position	1 Input	1 Analog Output	MDT, SSI, Analog, Quadrature	MA1, MA2, AA1, AA2
Velocity	1 Analog Input	1 Analog Output	Analog	AA1, AA2

Pressure	1 Analog Input	1 Analog Output	Analog Pressure	Any above + AP1 or AP2
Force	1 or 2 Analog Inputs	1 Analog Output	Analog Pressure	Any above + AP1 or AP2
Position-Pressure	1 Input + 1 Analog Input	1 Analog Output	Adding Pressure or Force requires an additional analog pressure input	Adding Pressure or Force requires an AP2 expansion module
Position-Force	1 Input + 1 or 2 Analog Inputs	1 Analog Output		
Velocity-Pressure	2 Analog Inputs	1 Analog Output		
Velocity-Force	2 or 3 Analog Inputs	1 Analog Output		

Note:
Force control axes require 1 input if a load cell is used, and require 2 inputs for differential force on a hydraulic cylinder.
Quadrature modules are currently not available on the RMC70.

Reference Axes

A reference axis does not have a Control output. It only has an input for transducer feedback. A reference axis is not capable of controlling a system. Reference axes are commonly used in gearing and synchronization applications. The A and AP Expansion Modules contain reference axes.

Reference Axis Types	Required Inputs	Transducer Types	Possible RMC70 Modules
Position	1 Input	MDT, SSI, Analog, Quadrature	A1, A2
Velocity	1 Analog Input	Analog	A1, A2
Pressure	1 Analog Input	Analog	AP1 or AP2
Force	1 or 2 Analog Inputs	Analog	AP1 or AP2
Virtual	none		

2. Starting Up the RMC70

2.1. RMC70 Start-up Procedure

The following step-by-step procedure will help you get your system up and running quickly. For a more detailed start-up procedure, use the Startup Guide that came with the RMC70.

> **TIP:**
> Deltas position/pressure simulators provide a simple way to test your program before connecting the controller to a real system.

1. Provide Power to the RMC Module
    a. Before providing power to the RMC for the first time, disconnect all other wiring from the module.
    b. Connect the power supply common to the **PS Return** pin on the RMC70 base module. The power supply should be rated for 500 mA.
    c. Connnect +24 volts to the **+24 Vdc PS** pin.
    d. For specifics on wiring, see the wiring topic. For details on the RMC power requirements, see RMC70 Specifications.

2. Connect to Controller in RMC70Tools
    a. Connect a Null-Modem cable with DB-9 connectors from a serial port on your computer to the RMC70.
    b. Open RMC70Tools.
    c. On the **File** menu, click **New Project**. In the **New Project Wizard**, fill in the fields and click **Finish**.
    d. In the **New Controller Wizard**, type a **Controller Name** and choose **Automatically Detect the Controller Information**.
    e. Click **Next**.
    f. Select the desired serial port, verify that the requirements in the **Note** are met, and click **Next**.
    g. You may need to wait while RMC70Tools connects to the controller. Once it has connected, verify that the information is correct and click **Finish**.

3. Connect a Feedback Device
    a. **Important:** Turn off power to the RMC70 and the feedback device before connecting any wires!
    b. For each axis you wish to connect a feedback device to, wire it to the RMC70 according to the instructions in the Wiring topic.
    c. After wiring, re-apply power to the RMC70 and the feedback device.
    d. In RMC70Tools, in the **Project** pane, click the **RMC70** controller.
    e. On the RMC70Tools toolbar, click the **Go Online** button (). Verify that the red circle around the RMC70 icon in the **Project** pane disappears, indicating that it is online with the RMC70.

f. The RMC70 must be properly configured before it will communicate with the transducer(s). Refer to the procedure for your module and your transducer type:

- MA Module - MDT transducer

  i. On the RMC70Tools toolbar, click the **Axis Tools** button ( ).

  ii. In the **Axis Parameters** pane, on the **Setup** tab, under the **Primary Control Setup**, in the **Feedback Type** register, select **MDT**.

  iii. In the **MDT Type** register, select the type of MDT transducer you have. This information should be available on the MDT datasheet. The options are:

  - Start/Stop Rising Edge
  - Start/Stop Falling Edge
  - Pulse-Width Modulated

  iv. To apply the changes to the RMC70, click the **Download** button ( ) or press Ctrl+D.

- MA Module - SSI transducer

  i. On the RMC70Tools toolbar, click the **Axis Tools** button ( ).

  ii. In the **Axis Parameters** pane, on the **Setup** tab, under the **Primary Control Setup**, in the **Feedback Type** register, select **SSI**.

  iii. From the information in your SSI transducer data sheet, enter the correct value for each of these registers:

  - **SSI Format** - Binary or Gray
  - **SSI Data bits** - (e.g. 24)

  ii. To apply the changes to the RMC70, click the **Download** button ( ) or press Ctrl+D.

- AA Module - Voltage or Current transducer

  i. On the RMC70Tools toolbar, click the **Axis Tools** button ( ).

  ii. In the **Axis Parameters** pane, on the **Setup** tab, under the **Primary Control Setup**, in the **Input Type** register, select **Voltage or Current**.

  iii. To apply the changes to the RMC70, click the **Download** button ( ) or press Ctrl+D.

b. On the RMC70Tools toolbar, click the **Axis Tools** button ( ).

c. In the **Axis Tools**, in the **Axis Status Registers** pane, on the **Basic Position** tab, look at the **Actual Position** register. It may be changing slightly.

d. Move the axis manually and look for a corresponding change in the **Actual Position** register. If it does not change, recheck the wiring, verify that the **Primary Control Setup** registers in the **Axis Parameters**

pane are correct, and check for changing **Actual Position** again. When it is working correctly, proceed to the next step.

> **Note:**
> It is important that the transducer is connected and is working properly before continuing the Start-up procedure.

e. On the **Controller** menu, click **Update Flash**. This stores your changes in the RMC70 even in the event of a power outage.

f. Press Ctrl+S to save the project.

4. Connect an Actuator

> **Note:**
> Read this section completely *before* executing any commands on the RMC70.

a. **Important:** Turn off power to the RMC70 and the actuator before connecting any wires!

b. For each axis you wish to connect an actuator to, wire it to the RMC70 according to the Wiring topic.

c. After wiring, re-apply power to the RMC70, the feedback device, and the actuator.

> **Note:**
> To test the actuator, you will supply a Control Output voltage from the RMC70 to the actuator. Before doing this, make sure that the axis may safely move in either direction!

d. In RMC70Tools, in the **Project** pane, click the **RMC70** controller.

e. On the RMC70Tools toolbar, click the **Go Online** button (). Verify that the red circle around the RMC70 icon in the **Project** pane disappears, indicating that it is online with the RMC70.

f. On the RMC70Tools toolbar, click the **Axis Tools** button ().

g. In the **Axis Parameters** pane, on the **All** tab, expand the **Simulate** section and verify that **Simulate Mode** is cleared. If you make changes, you must click the Download button () or press Ctrl+D to apply the changes to the RMC70.

h. On the RMC70Tools toolbar, click the **RUN Mode** button () to put the RMC70 in RUN mode.

i. In the **Axis Tools**, in the **Axis Status Registers** pane, on the **Basic Position** tab,

verify that the **Control Output** for the axis is 0 (zero).

j. Turn on power to the motor or hydraulics for the axis being set up. Note that after starting, it is normal for the axis to drift slowly.

k. On the **Tools** menu, click **Command Tool**. The following steps will ask you to issue commands from this tool. The steps to issue a command are as follows (do not issue a command yet):

  - In the Command Tool, in the desired axis, double-click the **Cmd** box and choose a command.
  - Enter the values for the command parameters, if there are any.
  - Click **Send Command** or press Alt+S.

l. In this step, you will apply 0.1 V to the axis Control Output:

Caution: Use the Direct Drive command with extreme caution! It disables the safety features of the RMC!

Note:
If you need to quickly halt the axis, click the Fault Controller button ( ) or press Ctrl+K. After doing so, you must click the **RUN Mode** button ( ) to put the RMC70 back into RUN mode.

  - Double-click the **Cmd** box, type "D" and choose the **Direct Drive (9)** command.
  - Set the **Output** to 0.1 V.
  - Set the **Ramp Rate** to 100 V/s.
  - Click **Send Command** or press Alt+S.
  - Enter 0 in the **Output** box. This will allow you to quickly issue a Direct Drive Command with 0 drive in the next steps.

m. Verify that the **Control Output** register for the axis is at 0.1 V.

n. Observe the **Actual Position** register (on the **Basic Position** tab) and note whether it is increasing or decreasing. One of three things could have happened at this point:

  - **The actuator moved and the Actual Position increased.**
    In this case you are ready to go

to the next section and set the Scale.

- **The actuator moved and the Actual Position decreased.**
  In this case, first verify that the wiring is correct. If it is not, fix it and repeat the process. If it is, invert the Output Polarity to reverse the direction:
  i. In the **Axis Parameters** pane, select the **Setup** tab and expand the **Primary Control Setup** section.
  ii. Double-click the **Invert Output Polarity** register in the axis you are using. The **Invert Output Polarity** box should now have a checkmark in it.
  iii. Click the **Download** button () or press Ctrl+D to apply the changes to the controller.
  iv. Repeat the process again: Issue the Direct Drive command again, observe the Actual Position, and see which of the three things happened.
- **The actuator does not move.**
  In this case, you may need to increase the Control Output. First, verify the following:

i. Check that the physical Control Output voltage really is 0.1 volts.
ii. Check that the actuator is enabled.

If these items check out fine, repeat the process above with a larger **Output**. Some systems may require much more than 0.1 volt to move.

> **Note:**
> It is essential that the Actual Position increase when a positive output voltage is specified with Direct Drive command. If this condition is not met, you will not be able to control the axis in closed loop.

b. In the Command Tool, enter 0 in the **Output** box and click **Send Command**, or press Alt + S, to turn the **Control Output** voltage off.
c. In the Command tool, enter -0.1 in the **Output** box and click **Send Command**, or press Alt + S. The **Control Output** register should change

to -0.1 and the axis should move in the opposite direction.

Once the actuator moves fine, you are ready to go to the next step.

5. Set the Scale and Offset

   The **Scale** and **Offset** parameters convert the **Counts** from the transducer into meaningful measurement units. In order to do so, the **Scale** and **Offset** must first be configured correctly.

   To set the Scale and Offset:

   a. In the **Axis Tools**, in the **Axes Parameters** pane, select the **Setup** tab. Note that the first two registers under the **Primary Control Setup** section are **Position Scale** and **Position Offset**. These are the parameters you will configure.
   b. To calculate what the scale and offset should be, you will need to refer to the Scaling topic.
   c. When you have determined what value the scale or offset parameter should be, type the new value in the cell in the **Axis Parameters** pane. Then click the **Download** button () or press Ctrl+D to apply the changes to the controller.
   d. On the **Controller** menu, click **Update Flash**. This stores your changes in the RMC70 even in the event of a power outage.
   e. Press Ctrl+S to save the project.

6. Tune Each Axis

   In order to control an axis in closed-loop control, it must first be tuned. Refer to the online help for the tuning procedure:

   a. In the **Axis Tools**, in **Axes Parameters** pane, select the **Tune** tab. You will set all the Gain and Feed Forward registers under the Position/Velocity Tuning section during tuning.
   b. Follow the instruction in the Tuning topic to tune the system.

7. Set up and Configure the Communications

   If your RMC70 will be communicating with an external controller, such as a PLC, PC, or other device, you must set up the communication. Refer to one or more of the following topics for details on

configuring the communication type of your RMC70 controller:

- PROFIBUS-DP
- Serial Overview

8. Save Your Configuration Settings

   Make sure to save your settings! This can be done in the following ways:

   - Save your RMC70Tools project. Before doing this, make sure the project data is the same as the controller data. To save the project, on the File menu, click **Save**.
   - On the **Controller** menu, click **Update Flash**.

2.2. Scaling

Scaling refers to converting the transducer feedback into meaningful units. The RMC70 uses the Scale and Offset parameters to convert the transducer Counts into measurement units (position, velocity, pressure, force). For example, the Voltage returned by an analog position transducer must be converted to positions in order to be useful for control. In order to correctly convert the transducer feedback to useful units, you must calculate the Scale and Offset parameters.

Scale and Offset Parameters

The following parameters converting the transducer feedback Counts into meaningful units. Each axis, whether a control axis or reference axis, has these parameters:

- **Scale**
  Defines the number of Actual units (position, velocity, pressure or force) per count or volt returned from the transducer.
- **Invert Feedback Polarity**
  Defines which direction the Actual units increase relative to the counts. This parameter allows the transducer to be mounted in either direction.
- **Offset**
  Moves the zero point of the Actual units to where the user wants it.

Calculating the Scale and Offset

The method of calculating the scale and offset parameters depends on the transducer type. See the scaling topic for the module and transducer you are using:

MA Module: MDT Scaling, SSI Scaling

AA Module: Analog Scaling

AP Module: Analog Scaling

2.3. Tuning

2.3.1. Tuning Overview

Once your system is set up and ready for use, it must be tuned in order to control it. Tuning is the process of adjusting the tuning parameters for optimum control of the system. The better tuned a system is, the closer the actual movement follows the desired path of movement.

Tuning Procedures

Tuning procedures differ depending on the type of system. Please read the **Tuning Guidelines** below before continuing to any of the tuning procedures. Click the following links for suggested tuning procedures:

- Tuning a Hydraulic Position Axis or Motor in Velocity Mode

Tuning Guidelines

Keep the following guidelines in mind throughout the tuning procedure. There is no substitute for experience when tuning an axis. The procedures offer some guidelines, tips, and suggestions for tuning your system. While the steps will work for many systems, they may not be the best for a particular system.

- **Reiterate these Steps**
  The tuning procedure is a reiteration of the following general steps. Use these steps throughout the tuning procedure:
  1. Make a move, typically using the Move Absolute (20) command. Use the Command Tool to issue the command.
  2. View the plotted move using the Plot Manager. Viewing the plot will help you determine which parameters must be changed.
  3. Change a parameter. Use the Axes Parameters pane in the Axis Tools to edit the parameters. After changing the value in the Axes Parameter Editor, you must click the download button  or press Ctrl+D to apply the changes to the controller.
  4. Repeat these steps using the same move until the parameter is at the desired value. See the Command Tool topic for details on using Shortcut commands to repeat moves.

- **Long, Slow Moves**
  Begin the tuning procedure with long, slow moves and low Accelerations. This will prevent you from losing control of and potentially damaging the system.
- **Use Small Plot Sample Interval**
  To obtain the greatest tuning precision, use the smallest Plot Sample Interval possible. This will allow you to see precisely how the system is responding.
- **Set Auto Stops**
  You may want to turn off some of the Auto Stop bits. The Auto Stop cause the axis to halt if an error occurs. In the initial stages of tuning, a Following Error or other error may occur, causing an undesired halt. Setting these bits to "Status Only" will make the RMC ignore the errors so you can tune the axis. Once you gain sufficient control of the axis, set the Auto Stop bits to halt. Turning off the Auto Stops may not be possible on some systems because of safety concerns.
- **Update Flash and Save the Project**
  When editing the parameters in the Project tag, you must download them to apply the changes to the RMC70. However, they are not updated in the RMC's Flash until you issue an Update Flash command. This will save the parameters even if power is disconnected. To save the RMC's parameters to the project, Upload the parameters and then save the project.

2.3.2. Tuning a Position Axis

The following procedure may be used to tune many position axes, hydraulic axes and motors in velocity mode. Please read the Tuning Overview topic before following this procedure. There is no substitute for experience when tuning an axis. This procedure offers some guidelines, tips, and suggestions for tuning your system. While these steps will work for some systems, they may not be the best for a particular system.

Tuning Procedure

1. **Test Wiring and Polarity**

   This step is for verifying that the system wiring and setup is correct before doing any closed loop control.

   - In the Command Tool, issue a Direct Drive (9) command to the axis. Use small **Output** value, such as 0.050-0.150 V. Use a **Ramp Rate** of 100. If the axis does not move, increase the **Output** until the axis begins to move.

   DANGER: The Direct Drive (9) command disables the safety features on the RMC! Use this command carefully!

   - A positive drive should yield increasing counts. If it does not, do the following:
     i. On the RMC70Tools toolbar, click the **Axis Tools** button ( ).
     ii. In the Axes Parameters Pane, on the **Setup** tab, toggle the Invert Output Polarity parameter.
   - Issue a Direct Drive (9) command again with a negative drive. This should yield decreasing counts.
   - Before continuing, verify that all the Gains and Feed Forwards are set to zero:
     i. In the **Axis Tools** button ( ), in the Axes Parameters Pane, on the **Tune** tab, set all the Gains and Feed Forwards to 0.
     ii. Click the **Download** button ( ) to apply the changes to the RMC70.

2. **Check the Deadband**

   If the axis exhibits a deadband, you may need to use the Deadband parameters.

   Check whether the system exhibits a deadband:

   a. Give increasing amounts of **Output** to the axis with the Direct Drive (9) command until the system starts to move.
   b. The value of **Output** at which the system starts to move is your deadband. If this value is approximately 0.4 V or greater, you should probably use the deadband parameters. If it is less, it is left to the discretion of the designer.

   Set the deadband parameters:

   If you found that your system has a deadband, set the deadband parameters in the following manner:

   c. On the RMC70Tools toolbar, click the **Axis Tools** button ( ).
   d. In the Axes Parameters Pane, on the **All** tab, expand the **Output** Section.
   e. Set the Output Deadband parameter to the value of your deadband.
   f. Set the Deadband Tolerance to a small value.
   g. Click the **Download** button ( ) to apply the changes to the RMC70.

2. **Adjust the Directional Gain Ratio**

   Some axes have different system gains depending on the direction of travel. The **Directional Gain Ratio** ratios the Gains and Feed Forwards to account for the different system gains.

> **Note:**
> Skip this step if you are not controlling a hydraulic cylinder.

To adjust the Directional Gain Ratio:

- Use the Open Loop Rate (10) command to move the axis in the extend direction. Use a value of **Requested Drive**, for example 1.0 V, that will move the axis at a reasonable speed. Make sure the cylinder reaches a constant speed. Be prepared to stop the axis by typing Ctrl+K (Disable Output (3) command) to keep it from running into the end of the cylinder. Record the constant **Actual Velocity** that the cylinder reached.
- Repeat the previous step for the opposite direction (use a negative **Requested Drive**).
- Using the following equation, calculate the **Directional Gain Ratio** parameter:

  Directional Gain Ratio = (Velocity in Negative Direction) / (Velocity in Positive Direction)
- In the **Axis Tools,** in the Axes Parameters Pane, on the **Tune** tab, enter the value you just calculated in the **Directional Gain Ratio** parameter.
- Click the **Download** button ( ) to apply the changes to the RMC70.

4. **Adjust the Output Bias**

   Some systems may drift significantly when the **Control Output** is at zero volts, which may adversely affect control. Use the **Output Bias** parameter to adjust the output such that the axis does not move when you issue an Open Loop command with zero volts **Control Output**. The RMC70 always adds the **Output Bias** to its **Control Output**.

   Check if your system need Output Bias:

   a. Issue an Open Loop Rate (10) command to the axis. Use a **Requested Output** of zero and a **Ramp Rate** of 100.
   b. If the axis moves significantly, you need to set the **Output Bias**.

   Set the Output Bias:

   c. In the **Axis Tools,** in the Axes Parameters Pane, on the **Tune** tab, enter a small number in the **Output Bias** parameter, e.g. 0.05 V or -0.05 V.
   d. Keep increasing (or decreasing) the number until the axis stands still.
   e. Click the **Download** button ( ) to apply the changes to the RMC70.

5. **Adjust the Proportional Gain**

The Proportional Gain must be adjusted to gain some control over the system for continuing the tuning procedure. You will fine-tune it later.

- Slowly increase the Proportional Gain as you make moves. When the Actual Position gets close to the Command Position reasonably quickly, continue to the next step. If the system begins to oscillate, decrease the gain. In this step, do not expect the Actual Position to track the Target Position very well during the move.
- Click the **Download** button ( ) to apply the changes to the RMC70.

6. **Adjust the Velocity Feed Forward**

   In many systems the Velocity Feed Forward parameter is the most important parameter for position tracking during a move. To adjust the Velocity Feed Forward:

   - Make long slow moves in both directions. Adjust the Velocity Feed Forward until the axis tracks within 10% in both directions. If it Tracks better in one direction than the other, go back to step 3 and adjust the Directional Gain Ratio.
   - Click the **Download** button ( ) to apply the changes to the RMC70.

7. **Adjust the Acceleration Feed Forward**

   The Acceleration Feed Forward parameter is particularly useful for systems moving large masses with relatively small cylinders. Such systems often have a delay before the start of movement. The Acceleration Feed Forward terms can help compensate for this delay.

   - Look for following errors during acceleration and deceleration. Increase the Acceleration Feed Forward parameter until the errors disappear.
   - Click the **Download** button ( ) to apply the changes to the RMC70.

8. **Readjust the Proportional Gain**

   Proportional Gain affects the responsiveness of the system. Low gains make the system sluggish and unresponsive. Gains that are too high make the axis oscillate or vibrate.

   - Slowly increase the gain. When you see a tendency to oscillate as the axis moves or stops, reduce the gain by 10 to 30 percent.
   - At this point, if you have gained sufficient control of the system, you may want to increase the speed, accel and decel of your moves and further adjust the proportional gain. A value of proportional gain that may

seem good at low speeds and accels may not work at higher speeds.

- Click the **Download** button ( ) to apply the changes to the RMC70.

9. **Adjust the Integral Gain**

   Many hydraulic systems do not require a large Integral Gain. However, it is usually desirable to have some Integral Gain to help compensate for valve null drift or changes in system dynamics. Some systems may require larger Integral Gain, in particular if they are moving a large mass or are nonlinear. Too much Integral Gain will cause oscillations and overshoot. The Integral Gain is helpful for getting into position and for tracking during long, slow moves. It will not significantly affect tracking during short, fast moves.

   - Click the **Download** button ( ) to apply the changes to the RMC70.

10. **Adjust the Differential Gain**

    Differential Gain may greatly enhance performance on many hydraulic systems. It is used mainly on systems that have a tendency to oscillate. This happens when heavy loads are moved with relatively small cylinders. Differential Gain will tend to dampen out oscillations and help the axis track during acceleration and deceleration. This will positively affect short, fast moves. A disadvantage of Differential Gain is that it amplifies position measurement noise. If there is too much noise or the gain is too high, this can cause the system to chatter or oscillate.

    - Increase the Differential Gain. It may help the system track better. If it starts oscillating or chattering, decrease the gain.
    - If the drive output during the constant velocity portion of the move is smooth, the Differential Gain is perhaps not set high enough.
    - When the Differential Gain is properly adjusted, the drive output may look "fuzzy." This indicates that the drive is responding to the minute errors of the axis. Not all systems allow the differential gain to be set high enough for the drive to be "fuzzy".

    > **Note:**
    > If you use Differential Gain, you may be able to increase the Proportional Gain somewhat without making the system oscillate.

    - Click the **Download** button ( ) to apply the changes to the RMC70.

11. **Increase System Speed**

Gradually increase the Speed and Acceleration values while making long moves. Look for following errors, overshoot, or oscillations.

- If an Output Saturated error occurs, there is not enough drive capacity to drive the axis at the requested Speed or Acceleration. Should this occur, reduce the Speed and/or Acceleration and Deceleration.
- If a Following Error occurs during acceleration and deceleration and adjusting the Gains and Acceleration Feed Forward does not help, the Acceleration and Deceleration ramps are too steep for the response of the system.
- If the actual position lags or leads the target position during the entire constant velocity section of the move, adjust the Feed Forwards.
- Should the system seem a little sloppy, try increasing the Proportional Gain.
- If the Control Output never gets very high, the gains can probably be increased for better control. If the Control Output is too high, or an overdrive error occurs, the system is not capable of performing the requested move. The Speed, and/or Accelerations may need to be decreased.
- If the system vibrates while in position, the gains may be too high, or the Dead Band value may need to be increased. However, if the oscillation is not caused by a deadband in the system, adjusting the Dead Band value will not help! A rule of thumb is to set the Dead Band Eliminator value to half of the peak-to-peak oscillation of the drive output while in position.
- The final tuning of the system should be made at the speed of intended operation.

12. Save your Settings to Flash

To retain your settings in the RMC70 in the event of a power loss, you must save the settings to Flash:

- On the **Controller** menu, click **Update Flash**.

3. Control Features

3.1. RMC70 Control Features

The RCM70 motion controller provides a host of features to successfully control any motion application. Browse the sections below for descriptions and links to the many features. For communication features, see Communications Overview.

Control

Types

Open Loop Control

Closed Loop Control

Controlled Quantities

Position

Velocity

Pressure/Force

Dual-Loop Control

Position-Pressure/Force

Position-Velocity

Motion Types

Point-to-Point (Absolute and Relative)

Coordinated Moves

SuperImposed Moves

Gearing

Splines, Camming

Motion Safety Features

Halts

Auto Stops

Enable Outputs

Fault Input

Feedback Types

Transducer Interfaces

MDT

SSI

Analog (+/- 10V, 4-20mA)

Measured Quantities

Position

Velocity

Pressure

Force (Single-Input and Dual-Input)

Orientations

Linear/Rotary

Absolute/Incremental

Reference Axes

Analog and Quadrature Reference

Filtering

Control Output

Servo Output (Voltage)

Programming

Pre-programmed Commands

User Program Overview

Variable Table

PreScan Table

Diagnostic Tools

Plots

Event Log

Status Registers

Miscellaneous

Digital I/O

3.2. Halts

3.2.1. Halts Overview

Halts stop motion on an axis. Halts will also stop any Tasks that are running, if the Programming Properties are set to do so. The RMC70 has four types, or levels, of halts to safely stop the axis in various circumstances. These halts are well-suited for use when error conditions occur. The RMC70, by default, triggers a halt when any Error Bits turns on.

Halt Types

The RMC70 has four types of halts:

- **External Halt**
  This halt only sets External Halt status bit. This halt type is intended to signal the PLC to initiate its fault handling.
- **Closed Loop Halt**
  The axis stays in closed loop and ramps down the Target Velocity to zero at the rate specified by the Closed Loop Halt Deceleration parameter. If an axis is in Open Loop, this halt will automatically be promoted to an Open Loop Halt.
- **Open Loop Halt**
  The axis will be put in open loop and the Control Output will be ramped down to the value of the Output Bias parameter using the Open Loop Halt Ramp parameter.
- **Open Loop Halt with Disable Output**
  This halt is the same as the Open Loop Halt but it also turns off the Enable Output when the Control Output reaches zero.

Halt Actions

When a halt occurs on an axis, the RMC70 takes the following actions:

- The axis begins the halt, as specified above.
- When a halt other than an External Halt occurs on an axis, the Halted status bit is set and the axis is said to be halted. If an External halt occurs, the External Halt status bit is set instead.
- If the axis is part of a Halt Group, it starts the same level of halt on all axes in the group. See the Halt Group Number topic for more details.
- The RMC70 immediately stops all Tasks by default. This setting can be changed on the Programming Properties dialog.

> **Caution:**
> If you disable this feature, User Programs may still be running after a halt occurs and may cause motion on the axis. Make sure you handle the halt condition safely. One method is to create a User Program to handle the halt condition and use the PreScan table to start the User Program when the Halt bit turns on.

Triggering a Halt

Halts can then be triggered in two ways:

- **Issue a Command**
  By issuing commands (except External Halt). See each halt command topic for details.

Halt	Command
Closed Loop Halt	Open Loop Halt (2)
Open Loop Halt	Closed Loop Halt (1)
Open Loop Halt with Disable Output	Disable Output (3)

- **Auto Stop**
  Auto Stops automatically trigger a halt when an Error Bit is set. The Auto Stops can be configured to trigger any level of halt, or Status Only, for each individual Error bit. See the Auto Stops topic for details.

3.2.2. External Halt

The External Halt is one of the four types of RMC70 Halts. Use the External Halt to signal to a master controller, such as a PLC, that an axis should be halted. The External Halt does not affect motion in any way.

When an External Halt occurs, it takes the following steps:

- The External Halt status bit is set.
- The action specified by the axis' Halt Step Action parameter is taken. The default Halt Step Action is to do nothing, but it can be set to start or stop a Task. For more details, see the Halt Step Action topic.
- If the axis is part of a Halt Group, it starts an External Halt on all axes in the group.

Triggering an External Halt

An External Halt can only be triggered via Auto Stops.

After a Halt has Occurred

If the halt was caused by an Auto Stop, you should first make sure the error condition that caused it has been resolved before continuing. Once it has been fixed, you can clear the External Halt status bit by issuing the Clear Faults (4) command. Issuing a valid motion command will also clear the Halted status bit.

3.2.3. Closed Loop Halt

The Closed Loop Halt is one of the four types of RMC70 Halts. Use the Closed Loop Halt to halt the axis while remaining in closed loop control.

When a Closed Loop Halt occurs, it takes the following steps:

- The axis stays in closed loop and starts ramping down the Target Velocity to zero at the rate specified by the Closed Loop Halt Deceleration parameter. If the axis is in Open Loop when this halt occurs, this halt will start an Open Loop Halt instead.

> **Note:**
> It is the *Target* Velocity that is ramped down. If the Actual Position is lagging behind the Target Position, it will not stop until it reaches the Target Position. If you need the Actual Position to stop immediately, use an Open Loop Halt or Stop command instead.
> **Note:**
> The deceleration specified by the Closed Loop Halt Deceleration is the *average* deceleration. The instantaneous deceleration may exceed this value.

- The Halted status bit is set and the axis is said to be in the halted state.
- The action specified by the axis' Halt Step Action parameter is taken. The default Halt Step Action is to do nothing, but it can be set to start or stop a Task. For more details, see the Halt Step Action topic.
- If the axis is part of a Halt Group, it starts a Closed Loop Halt on all axes in the group.

Triggering a Closed Loop Halt

A Closed Loop Halt can be triggered in two ways:

- By issuing the Closed Loop Halt (1) command.
- Via Auto Stops.

After a Halt has Occurred

If the halt was caused by an Auto Stop, you should first make sure the error condition that caused it has been resolved before continuing. Once it has been fixed, you can clear

the Halted status bit by issuing the Clear Faults (4) command. Issuing a valid motion command will also clear the Halted status bit.

Why Bother?

This halt is useful when you want to stop the axis but remain in closed loop control. If you have to stop the axis because it is vibrating, use the Open Loop Halt instead, since the axis will probably keep vibrating or oscillating as long as it is in closed loop control.

The Auto Stops can be set up to cause this halt when an error bit turns on.

3.2.4. Open Loop Halt

The Open Loop Halt is one of the four types of RMC70 Halts. Use the Open Loop Halt to halt the axis in open loop control.

When an Open Loop Halt occurs, it takes the following steps:

- It puts the axis in open loop and ramps down the Control Output to zero using the Open Loop Halt Ramp parameter.
- The Halted status bit is set and the axis is said to be in the halted state.
- The action specified by the axis' Halt Step Action parameter is taken. The default Halt Step Action is to do nothing, but it can be set to start or stop a Task. For more details, see the Halt Step Action topic.
- If the axis is part of a Halt Group, it starts an Open Loop Halt on all axes in the group.

Triggering an Open Loop Halt

An Open Loop Halt can be triggered in two ways:

- By issuing the Open Loop Halt (2) command.
- Via Auto Stops.

After a Halt has Occurred

If the halt was caused by an Auto Stop, you should first make sure the error condition that caused it has been resolved before continuing. Once it has been fixed, you can clear the Halted status bit by issuing the Clear Faults (4) command. Issuing a valid motion command will also clear the Halted status bit.

Why Bother?

This halt is useful when you want to set the Control Output to zero because of a potentially dangerous error. The Open Loop Halt Ramp parameter is used to avoid an abrupt (and potentially damaging) stop. If you just want to stop the axis but still have closed loop control over it, use the Closed Loop Halt.

The Auto Stops can be set up to cause this halt when an error bit turns on.

3.2.5. Open Loop with Disable Output Halt

The Open Loop with Disable Output Halt is one of the four types of RMC70 Halts. Use this halt to stop the axis in open loop control and turn off the Enable output.

This halt is similar to the Open Loop Halt but with the following differences:

- This halt also turns off the Enable output when the Control Output reaches zero.
- To initiate this halt using a command, you must issue the Disable Output (3) command.

See the Open Loop Halt topic for details.

3.3. Control Modes

3.3.1. Control Modes Overview

This topic explains basic motion control modes and also feedback types that the RMC70 can interface with.

Motion Control Modes

Motion control can be classified into two types: Open Loop and Closed Loop. The RMC70 has motion commands of both types.

Open Loop Control

Open Loop Control is the simplest form of control. A signal is given to a system, for example, 2 volts of Control Output to a valve, causing a cylinder to extend at some approximate speed. In open loop control, there is no way of commanding the system to go exactly at a specific speed or go to an exact position. See the Open Loop Control topic for more details.

Closed Loop Control

Closed Loop control uses feedback from the system being controlled. For example, a command is issued to go to 20 inches. The RMC70 computes a target (a motion path) to get to 20 inches. The controller (the RMC70) uses the feedback and the gains to compute every control loop the amount of Control Output that should be given to the valve to follow the profile. The system will automatically go to 20 inches. See the Closed Loop Control topic for more details.

The RMC70 uses a type of closed loop control called PFID. This type of control employs several gains to compute the Control Output. See the PFID topic for a full description.

RMC70 Feedback Control Types

The RMC70 supports closed loop control with several types of feedback, as long as the feedback type has a transducer compatible with the RMC70. The RMC70 supports many transducer types. In addition, any type of feedback can be controlled if the transducer is compatible with the RMC70.

Feedback Type	Required Transducer Type	Required Inputs	Required Outputs
Position	MDT (Magnetostrictive Displacement Transducer) or SSI (Serial Synchronous Interface)	1 Input	1 Analog Output
Velocity	Analog Voltage or Current (+/-10V or 4-20 mA)	1 Analog Input	1 Analog Output
Pressure	Analog Voltage or Current (+/-10V or 4-20 mA)	1 Analog Input	1 Analog Output
Force	Analog Voltage or Current (+/-10V or 4-20 mA)	1 or 2 Analog Inputs	1 Analog Output

The RMC70 also supports dual-loop control:

Feedback Type	Required Inputs	Required Outputs

Position-Pressure	1 Input + 1 Analog Input	1 Analog Output
Position-Force	1 Input + 1 or 2 Analog Inputs	1 Analog Output
Velocity-Pressure	1 Input + 1 Analog Input	1 Analog Output
Velocity-Force	1 Input + 1 or 2 Analog Inputs	1 Analog Output

3.3.2. Closed Loop Control

Closed Loop control uses feedback from the system being controlled. First, the RMC70 generates a Target (position or velocity) which specifies where the axis should move to. Then, the RMC calculates how much Control Output should be given to get to the Target. The RMC70 repeats this for every control loop time. All the closed-loop commands specify the target to be generated. The tuning parameters specify how to calculate the Control Output, which makes the axis follow the target.

The RMC70 uses a type of closed loop control called PFID. This type of control employs several gains to compute the Control Output. See the PFID topic for a full description.

Primary and Secondary Closed-Loop Control

Primary Control is the axis module, Secondary Control is the expansion module, which is pressure or force. These are both assigned to a single axis. An axis can be enter closed-loop pressure control whether it is in open-loop or closed-loop position or velocity control.

Closed Loop Commands

The RMC70 has the following closed-loop commands:

Motion

Move Absolute (20)

Move Relative (21)

Quick Move Absolute (15)

Quick Move Relative (16)

Time Move Absolute (23)

Time Move Relative (24)

Speed at Position (36)

Halting

Closed Loop Halt (1)

Hold Current Position (5)

Stop (6)

Pressure/Force

Enable/Disable Pressure Limit (40)

Ramp Pressure (41)

3.3.3. Open Loop Control

Open Loop Control is the simplest form of control. A signal is given to a system, for example, 2 volts of Control Output to a valve, causing a cylinder to extend at some approximate speed.

In open loop control, there is no way of commanding the system to go exactly at a specific speed or go to an exact position.

Sometimes, improperly tuned closed-loop control may cause a system to oscillate or exhibit other erratic behavior. Usually, issuing an open-loop command with 0 Requested Output will stop the system safely. See Auto Stops for details on halting a system.

Some of the RMC70 open-loop commands have a ramp rate parameter that specifies the rate at which the Control Output ramps up to the Requested Output. The ramp avoids jerking the system.

Open Loop Commands

The RMC70 has the following open-loop commands:

- Open Loop Rate (10)
  This is the basic open-loop command. It ramps the Control Output from the current value to the Requested Output at the rate specified by the Ramp Rate parameter. Use this command for normal open-loop moves.
- Direct Drive (9)
  This command is similar to the Open Loop Rate command, except it disables all the safety features of the RMC70. It is intended only for testing the Control Output. Use the Open Loop Rate command for all other purposes.
- Open Loop Absolute (11) and Open Loop Relative (12)
  These are specialty open loop commands. They specify the Control Output as a function of distance. As such, they do require position feedback on the axis.

Open Loop Halt (2)

Quick Move Absolute (15) and Quick Move Relative (16)

3.3.4. Pressure Limit

This topic describes how to perform pressure control with the RMC70. To perform position-pressure control, see the Position-Pressure Control topic.

Requirements

Pressure control requires one analog output and an analog pressure input. This requires the following RMC70 modules:

- Any Axis Module
- An AP2 Expansion Module

Setting Up an Axis for Pressure Control

Follow these steps to set up an axis for pressure control:

1. Wire the transducer.
2. Configure the Axis Parameters.

Controlling Pressure

After setting up the axis for pressure control, follow these steps to limit pressure:

1. Issue the Ramp Pressure (41) command to set the Command Pressure to the desired value.

> **Note:**
> The Ramp Pressure command must be issued before the Enable/Disable Pressure Limit command, or a Command Error will result.

2. Use Open Loop Rate (10) command to move the axis to a point where the pressure is below the Command Pressure.
3. Issue the Enable/Disable Pressure Limit (40) command.
4. Issue an Open Loop Rate (10) command to move the axis. When the motion begins to affect the pressure, the RMC70 will limit the motion such that the Actual Pressure does not exceed the Command Pressure.
5. To change the Command Pressure, issue the Ramp Pressure (41) command.
6. To make an axis go to a certain pressure, the axis must be commanded to move to a point at or beyond the point where the pressure limit is reached.

> **Note:**
> Pressure Limit mode may affect normal motion even when the pressure is very low. Therefore, if it is possible, do not enter Pressure Limit until you need to.

3.3.5. Position-Pressure Control

Many applications require position control and pressure control on the same axis. The RMC70 excels at limiting pressure while controlling position.

Requirements

Position-pressure control requires the following:

- Any Axis Module.
- An AP2 Expansion Module.

Setting Up an Axis for Position-Pressure Control

Follow these steps to set up an axis for position-pressure control:

1. Wire the transducers.
2. Configure the Axis Parameters.

Controlling Pressure

After setting up the axis for pressure control, follow these steps to limit pressure:

1. Issue the Ramp Pressure (41) command to set the Command Pressure to the desired value.

> **Note:**
> The Ramp Pressure command must be issued before the Enable/Disable Pressure Limit command, or a Command Error will result.

2. Issue the Enable/Disable Pressure Limit (40) command.
   The RMC70 will now allow normal motion to be executed on the axis. However, when the motion begins to affect the pressure, the RMC70 will limit the motion such that the Actual Pressure does not exceed the Command Pressure.
3. To change the Command Pressure, issue the Ramp Pressure (41) command.

3.3.6. PFID Control

The RMC70 employs a PFID control loop to provide closed-loop control of motion. PFID stands for:

- **P**roportional Gain
- **F**eed Forward (Velocity FF and Acceleration FF) Gain
- **I**ntegral Gain

- **D**ifferential Gain

The PFID loop employs these gains and the transducer feedback to generate a Control Output in volts to drive the valve, motor, or other actuator. The PFID always attempts to produce motion such that the axis moves toward the target position (or velocity, pressure or force).

The RMC70 motion commands generate a target profile. The PFID is responsible for keeping the axis as close to the target profile as possible.

Velocity and Torque Mode

Most actuators, together with their power source and/or drive electronics, can be classified in two types: *velocity mode* or *torque mode*. Which type it is affects the tuning procedure and how the actuator handles certain RMC70 commands.

Definition

The actuator type is defined by its response to the Control Output voltage applied by the RMC70:

- A velocity mode actuator produces a *speed* proportional to the Control Output.
- A torque mode actuator produces a *torque* or *force* proportional to the Control Output.

In practice, this means that if you issue an Open Loop Rate command with 5 volts Control Output to a velocity mode actuator, the actuator will move at a speed proportional to the 5 volts. If you issue the same Open Loop Rate command to a torque mode actuator, the actuator will provide a torque proportional to the 5 volts. The actuator speed will keep increasing until the torque is equal to the friction in the system. The final drive speed for a torque mode actuator is not necessarily proportional to the 5 volt output.

Note:
It is not recommended that you issue Open Loop commands to a torque mode system during normal operation, because you cannot predict the final speed. Use closed loop commands instead.

Examples

Velocity mode actuators:

- Hydraulic cylinder
- Motors with a velocity drive

Torque mode actuators:

- Motors without an inner velocity loop

Effect on Tuning

Whether an actuator is velocity or torque mode affects the tuning of a system in the following ways:

- **Damping**
  With a Control Output of 0, a velocity mode actuator is normally difficult to move manually. This means the system has significant damping. A torque mode actuator however, is relatively easy to move manually and has very little damping. When tuning a torque drive, some damping must be provided initially with the Differential gain. This is the primary difference between the tuning methods of velocity drives and torque drives.
- **Feed Forwards**
  On a velocity mode system, the Velocity Feed Forwards often provide most of the drive required to move the axis and may therefore be large. On a torque mode system, the Velocity Feed Forwards are basically only for overcoming friction and are often small. On torque mode systems, the Accel Feed Forwards do a lot of the work.

Effect on Commands

The Open Loop Absolute (11) and Open Loop Relative (12) commands are intended only for velocity drives.

3.4. Plots

3.4.1. Using Plots

The RMC70 provides very flexible plotting capabilities. Virtually any register in the RMC70 can be plotted, and multiple registers may be plotted simultaneously. The plot trigger allows events to easily be captured. Plots can be controlled and viewed with the Plot Manager in RMC70Tools, or with a PLC or other host controller by using the plot commands.

Use the Plot Manager to view plots in RMC70Tools. For advanced plotting features, see the Advanced Plot Capabilities topic.

How to plot a Move

The default RMC70 setting is to automatically trigger a plot when a motion command is issued. If you have not changed the plot settings from their defaults, follow these steps to view a plot of motion:

1. Issue a motion command to an axis.
2. In the Project Pane, expand the **Plots** folder under the current controller.
3. If you issued a plot to Axis 0, double-click Plot 0. If you issued a plot to Axis 1, double-click Plot 1.
4. The Plot Manager will open and the plot will be uploaded to the window.
5. For instructions on viewing and using plots in the Plot Manager, see the Plot Manager topic.

If you have changed the plot defaults, then the above steps may not work. Read the Advanced Plot Options topic for a full description of plotting in the RMC70.

Basic Plot Settings

One of the most basic plot settings is the **Plot Duration**. If the plot is not long enough to see an entire move, change this setting.

1. In the Project Pane, expand the desired controller and expand the **Plots** folder.
2. Right-click the icon of the desired plot and choose **Properties**. The **Plot Properties** dialog will appear.
3. Click the **General** tab.
4. In the **Plot Duration** box, enter the length of the plot. Changing the Plot Duration will affect the **Total Samples**, which has a maximum allowed value. If the **Total Samples** value becomes red, the Plot Duration is too long.

> **Tip:**
> If you need a longer Plot Duration, you can increase the **Sample Interval**. Then you can enter a longer **Plot Duration**.

5. Click **OK**.
6. Right-click the **Plots** folder in the **Project** pane and choose Download Plot Configurations.

Advanced Plot Options

For advanced plot options, such as allocating the plot memory, selecting data to plot, and triggering plots, see the Advanced Plot Capabilities topic.

3.4.2. Advanced Plot Capabilities

The RMC70 default plot settings are intended to be easy to use. This topic will introduce more advanced plot capabilities to assist advanced troubleshooting. For general information on plots, see the Plot Overview topic.

Plot Options

Allocating the Plot Memory

The RMC70 has a fixed amount of memory available for plots, that can be allocated as you wish. See the Configuring Plots topic for details on how to change the following settings:

- **Number of Plots**
  Specifies the total number of plots on the RMC70.
- **Data Sets per Plot**
  Specifies how many registers can be plotted in a plot.

How the memory space is allocated determines the maximum number of samples per plot, which in turn affects the maximum Plot Duration and Sample Interval of each Plot. If you want small sample times and/or long plots, you may have to choose a small number of plots and/or a small number of Data Sets.

Plot Duration and Sample Interval

You may specify the Plot Duration and Sample Interval for each plot. The Plot Duration specifies the plot length. The Sample Interval specifies the time between samples in a plot. Since the number of samples per plot is fixed (see Allocating the Plot Memory above), a longer Plot Duration will increase the minimum Sample Interval. Likewise, a shorter Sample Interval will decrease the maximum Plot Duration. The Sample Interval cannot be smaller than the Control Loop Time.

See the Configuring Plots topic for details on how to change these settings.

Data to Plot

You can plot any register in the RMC70. To find out how to add a register to a plot, see the Selecting Data to Plot topic.

Triggering Plots

You can manually trigger a plot or set it to trigger automatically on certain conditions. Triggering a plot also allows you to see data from before you triggered the plot and after you triggered the plot. See the Triggering Plots topic for more details.

Plot Commands

You can use the following commands to start, stop and trigger plots. Issue these commands like any other command. Some of these commands can also be issued directly from the Plot Manager. See each command topic for more details.

Plot Command	Function
Start Plot (100)	Starts a plot immediately.
Stop Plot (101)	Stops a plot immediately.
Trigger Plot (102)	Triggers a plot. See the Triggering Plots topic for more details.
Rearm Plot (103)	Re-arms a plot so that it can be triggered again.
Enable/Disable Plot Trigger (104)	Enables or disables the plot trigger.

3.4.3. Configuring Plots

To configure a plot, first allocate the plot memory, and then configure the individual plot.

Allocating Plots

The RMC70 has a limited memory space for plots. The user may choose how to allocate this space by selecting the number of plots and the number of data sets per plot. As the number of plots increases, the maximum number of data sets per plot decreases.

To change the Plot allocation, follow these steps:

1. Right-click the **Plots** folder under the desired controller in the Project Pane. Click **Properties**.
2. The **Number of Plots** and **Data Sets per Plot** values can be changed. The numbers in parentheses indicate the range of valid values.
3. Edit the desired items and click **OK**.
4. Right-click the **Plots** folder in the **Project** pane and choose Download Plot Configurations.

Configuring Plots

Each plot can be configured individually. To configure a plot, follow these steps:

1. In the Project Pane expand the desired controller and expand the **Plots** folder.
2. Right-click the icon of the desired plot and choose **Properties**. The **Plot Properties** dialog will appear.
3. Edit the desired items. The **Plot Properties** dialog has three tabs:

General Tab

This tab contains general settings:

Setting	Description
Name	Descriptive name for user reference.
Plot Duration	Specifies the plot length in seconds. If the **Total Samples** value becomes red when increasing this value, it is too large.
Sample Interval	Specifies the time between each sample. A smaller value will result in finer detail on the plot. A larger Sample Interval will allow for a longer **Plot Duration**
Total Samples	This non-editable value shows the total number of samples in the plot. This value is calculated from the selected **Plot Duration** and **Sample Interval**. The **Total Samples** must be within the range given in parentheses. Otherwise, it will be truncated to the maximum valid value.
Assigned to Axis	Associates the plot with an axis. If the plot is associated with an axis, the plot commands 100-104 can be used with a value of -1. The "Assigned Axis" option on the Trigger tab of the Plot Properties dialog may also be used.

Data Tab

The list in the Data tab displays the data items that will be plotted. Use the buttons to edit this list:

Button	Description
New	Add a new data item to the list.
Edit	Edit the selected data item on the list.
Delete	Remove a data item from the list.
Move Up	Move the selected data item up.
Move Down	Move the selected data item down.

Trigger Tab

Use this tab to set up the plot trigger. See the Plot Trigger topic for details.

4. Click **OK**.
5. Right-click the **Plots** folder in the **Project** pane and choose Download Plot Configurations.

3.4.4. Triggering Plots

Triggering a plot means capturing data from the past and the future and plotting. Triggering a plot allows you to see data from before you triggered the plot and after you triggered the plot. This is a valuable aid in troubleshooting. For example, if you trigger the plot to occur when a motion command is issued, then you can plot the data from before the command was issued until some time after it was issued. To specify how much of the plot data is from before the plot was triggered and how much is after the plot trigger, you must use the Trigger Position value.

Note:
Compare *triggering* a plot to *starting* a plot with the Start Plot (100) command. The Start Plot command starts plotting immediately and does not contain any past information. The Start Plot command cannot automatically start a plot like the plot trigger does.

Note:
You can entirely enable or disable the plot trigger by issuing the Enable/Disable Plot Trigger (104) command. A plot cannot be triggered unless the plot trigger is enabled.

Using the Trigger Position

The Trigger Position specifies how much of the plot data is from before the plot was triggered and how much is after the plot Trigger. The Trigger Position is a percentage from 0% to 100%. If the Trigger Position is set to 0%, the triggered plot will not contain any past data. If the Trigger Position is set to 100%, the triggered plot will contain only past data. To set the Trigger Position, see the **Changing the Trigger Settings** section below.

Example:

You have set the Plot Duration to 4 seconds and the Trigger Position to 25%. When you trigger a plot, the first 1 second of the plot will contain data from immediately before the trigger occurred, and the last 3 seconds will contain data from immediately after the trigger occurred.

How to Trigger a Plot

Note:
To trigger a plot, the trigger must first be armed. See the **Re-arming the Trigger** section below.

There are two ways of triggering a plot: manually and automatically:

- Automatic Trigger:

  Plots can be automatically triggered by certain conditions. Currently, a plot can be automatically triggered only by motion commands. If automatic triggering is enabled, a plot will trigger every time a motion command is issued.

  You must specify which axis the motion command must be issued to in order to trigger a plot. You may explicitly specify one of the axes or you may specify "assigned axis". Choosing "assigned axis" will cause the plot to trigger on motion commands that are issued to the axis you have assigned the plot to. See theAssigning Plots to Axes topic for details on assigning axes to plots.

  To set up automatic triggering, see the **Changing the Trigger Settings** section below.

> **Note:**
> The RMC70 default setting is to automatically trigger plots on motion commands.

- Manual Trigger:

  To manually trigger a plot, issue the Trigger Plot (102) command. You can also issue the Trigger Plot command from the Plot Manager by clicking the Trigger Plot  icon in the toolbar.

> **Note:**
> Even if you manually trigger a plot, the RMC70 may also automatically trigger plots because it is set to do so by default. To disable automatic triggering, see the **Changing the Trigger Settings** section below.

Re-arming the Trigger

Before triggering a plot, the trigger must first be *armed*. This means that the RMC70 is prepared to trigger when the Trigger Plot command is issued. If you issue the Trigger Plot command when the trigger is not armed, the plot will not trigger. When the RMC70 starts up, the trigger is armed and therefore prepared to trigger. After triggering a plot, the trigger is no longer armed and must be re-armed before you can trigger a plot again.

There two methods of re-arming the trigger:

- Automatically Re-arm:

  The plot trigger can be set to automatically re-arm after it triggers. This is the default RMC70 plot setting. After a plot triggers, it is ready to be immediately triggered again.

> **Note:**
> If you are issuing motion commands in rapid succession, the automatic re-arm feature may cause the plots to be cut short when the next plot is triggered. To get long plots, you should change the plot settings to *manual* re-arm. See the **Changing the Trigger Settings** section below.

- Manually Re-arm:

  To manually re-arm the trigger, issue the Rearm Plot (103) command. You can also issue the Re-arm Plot command from the Plot Manager by clicking the Re-arm Trigger  icon in the toolbar. To enable manual re-arming, see the **Changing the Trigger Settings** section below.

Changing the Trigger Settings

The trigger settings can be configured for each individual plot. To change the trigger settings for a plot, follow these steps:

1. In the Project Pane expand the desired controller and expand the **Plots** folder.
2. Right-click the icon of the desired plot and choose **Properties**. The **Plot Properties** dialog will appear.
3. Click the **Trigger** tab.
4. Make the desired changes and click **OK**.
5. If you are online with the controller, the changes will be downloaded to the controller immediately. If you are not online, you must download the changes later after you go online. To download the changes, in the Project pane, right-click **Plots** and click **Download Plot Configuration**.

3.4.5. Selecting Data to Plot

Any register in the RMC70 can be plotted. To select which registers will be plotted, use the following steps:

1. In the Project Pane expand the desired controller and expand the **Plots** folder.
2. Right-click the icon of the desired plot and choose **Properties**. The **Plot Properties** dialog will appear.
3. Click the **Data** tab.
4. The list specifies which registers will be plotted in the selected plot. Note that each plot has maximum number of registers that may be plotted. To add, change, or delete registers in the list, use the buttons to the right of the list:

Button	Description
New	Add a new register to the list.
Edit	Edit the selected register in the list.
Delete	Remove the selected register from the list.
Move Up	Move the selected item up.
Move Down	Move the selected item down.

5. Click **OK**.
6. Right-click the **Plots** folder in the **Project** pane and choose Download Plot Configurations.
7. The next time you start this plot, it will contain the registers you entered in the list.

3.4.6. Assigning Plots to Axes

You can assign a plot to an axis. Several plots can be assigned to a single axis. A plot does not have to be assigned to an axis.

Why Assign Axes?

Assigning plots to axes affects the following items:

- If you issue a plot command with the parameter **Plot Number** set to -1, the command will affect *all* plots that are assigned to the axis you issued the command to. For example, you can use this feature to start two plots at the same time.
- If you have selected Automatic Trigger on motion commands, you must specify which axis will cause the trigger. You can select a specific axis, or the axis the plot is *assigned* to.
  Example:
  If you select Automatic trigger on motion commands of the assigned axis, and the plot is assigned to Axis 1, then the plot will trigger whenever a motion command is issued to Axis 1.

To assign a plot to an axis

1. In the Project Pane expand the desired controller and expand the **Plots** folder.
2. Right-click the icon of the desired plot and choose **Properties**. The **Plot Properties** dialog will appear.
3. Click the **General** tab.
4. Select the axis in the **Assigned to Axis** box and click **OK**.
5. If you are online with the controller, the changes will be downloaded to the controller immediately. If you are not online, you must download the changes later after you go

online. To download the changes, in the Project pane, right-click **Plots** and click **Download Plot Configuration**.

Default Settings

The default, the RMC70 axis assignment settings for plots are:

- Plot 0 assigned to Axis 0
- Plot 1 assigned to Axis 1
- Any other plots are not assigned to an axis.

3.4.7. Using Plots with a Host Controller

RMC70 plots can be fully controlled and accessed with a PLC or other host controller. To start, stop, or trigger a plot, use the plot commands. The plot can be read from the RMC70 like any registers, but the plot must be finished before reading for the data to be meaningful.

Plot Commands

The RMC70 has several commands specifically for plots.

Start Plot (100)

Stop Plot (101)

Rearm Plot (103)

Trigger Plot (102)

Enable/Disable Plot Trigger (104)

Reading a Plot

To read a plot from the RMC70, first ensure that the plot has finished. Then read the registers in the plot area as you would any registers.

4. Programming

4.1. User Programs

4.1.1. User Programs Overview

User Programs carry out complex sequences of commands on the RMC70 without intervention from a PLC or other controller. This allows the RMC70 to respond to events within the control-loop time rather than the scan rate of a PLC. It also reduces the PLC programming required.

Each User Program can contain any number of steps. You can link to different programs simply by specifying the label to jump to. You can also start other User Programs from within a User Program.

Use variables to make User Programs flexible, and to easily influence them from a PLC. Perform math operations with the Expression (113) command, and make the programs readable by using tag names in the Command Parameters, Link Types, and in the Expression (113) command.

User Programs can be started by issuing a command, or from the PreScan Table.

There number of User Programs is only limited by the memory capacity of the RMC70. See the Verify Results pane to find how much memory the User Programs are using.

Tasks

User Programs run on **Tasks**. The RMC70 has up to 4 Tasks. One User Program can run per Task. Therefore, up to 4 User Programs can run simultaneously on the RMC70. To start a User Program, issue the Start Task (90) command. It starts the specified User Program in the specified Task. A Task can also be started from within a User Program or from the PreScan Table. For more details, see the Running User Programs topic.

Steps for Creating and Running a User Program

- Create a Program
- Verify the Program
- Download the Program
- Run the Program

Structure of User Programs

A User Program consists of multiple steps which execute one command on one or several axes. The series of steps are linked together in sequences. Each step has the following parts:

- **Command**
  A step can issue any RMC70 command.
- **Commanded Axes**
  The command may be issued to one or several axes simultaneously.
- **Link Type**
  Specifies when to jump to the next step and which step to jump to. For example, a link type can wait until the axis is in position before going to the next step, or wait a certain amount of time. You can also define a complex link condition by entering a mathematical expression. See the Link Type topic for details.

4.1.2. Creating User Programs

Follow these steps to create a User Program:

> **Note:**
> After you have created a User Program, you must verify and download it before running it on the RMC70.
> **Note:**
> In order to run User Programs or the PreScan Table, the RMC70 must be in RUN mode. See the RUN Mode and PROGRAM Mode topics for details.

1. **Open a User Program**
   - Open a new User Program
     - In the Project Pane, expand the desired controller, expand the **Programming** node, and right-click **User Programs**.
     - Click **New Program.**
     - Type the name of the program and click **Finish**.
   - Open an existing User Program

     In the Project pane, expand the desired controller, expand **Programming**, expand **User Programs**, and double-click the desired program.

2. **Add Steps**

   You may add steps to a User Program at any time.
   - Appending a Step

     To add a step to the end of the program:
     - Right-click anywhere in any step and click **Append Step**.
   - Inserting a Step

     To insert a step in a program:
     - Right-click anywhere in the step and click **Insert Step**.
     - The step will be inserted above the step you clicked on.

3. **Fill Out the Steps**

   You may edit the steps at any time. For each step in the User Program, do the following:

   a. Select a Command

      There are 2 methods of selecting a command:
      - Click the **Command** box and then click the Details button. Choose a command from the Command List and click **OK**. The commands are grouped by type in the hierarchical list to help you find the one you want.
      - Click the **Command** box and begin typing the command name or number. Use the arrow keys to select the command.

   b. Enter the Command Parameters

      If the command has any parameters, enter their values. You may enter a number, Tag name, or address.

      > **Tip:**
      > For help on the command's parameters, click the command box and press F1.

      - Click a parameter box.
      - To enter a **number**, type the number.
      - To enter an **address**, begin typing the address and then use the arrow keys to select the address.

- To enter a **Tag** name:

> **Note:**
> A Tag may be a Variable or any register in the RMC70.

  - Click the Details button [...] and choose the Tag from the list.
  - Or, start typing the Tag name and then use the arrow keys to select the address.

c. Select Commanded Axes

The Commanded Axes specifies which axes to issue the command to.

- Select the check box of each axis you wish to issue the command to. For example, selecting the Axis 0 and Axis 1 check boxes will make the command be issued to both axes.

> **Note:**
> If you do not select a Commanded Axis, the command will be issued to the same axis as the Start Task command was issued to.

d. Enter a Link Type

The Link Type specifies when the program will jump to another step, and which step it jumps to.

- Choose one of the Link type in the **Link Type** box:

  These link types jump to the next step:
  - End
  - Immediate
  - Delay
  - Wait For

  These link types jump to a step number or a step label:
  - Jump
  - Delay Jump
  - Conditional Jump

e. Add a Comment

If you wish, you may add a comment to the step. Comments help you keep track of what the step is for and what it does.

- Right-click anywhere in the step and click **Add Comment**.
- Type your comment and press Enter.

f. Add a Step Label

If you wish, you may add a label to the step. Labels can be used when specifying the step to jump to. See the labels topic for details on labels.

- Right-click anywhere in the step and click **Label Step**.
- Type the label name.
- Click **OK**.

4. Additional Editing

Do any additional editing:

- Delete a step
  - Right-click any empty space in the step and click Delete.
- Delete a comment
  - Right-click the comment and click **Delete Comment**.
- Hide a comment
  - Right-click the comment and click **Hide Comment**.

- Show a comment
    - Right-click the yellow bar at the top of the step and click **Show Comment**.

4. **Verify the Program**

   After you have created the program you must verify it before downloading it and running it. See the Verify topic for instructions.

4.1.3. Verifying User Programs

After you have created a User Program, it must be verified before downloading and running it in the RMC70. The verify process checks the program for errors that will keep it from running. It does not check for other errors, such as invalid command parameters or bad logic. If errors exist, the verify will fail.

To Verify the User Programs

1. In the Project Pane, expand the desired controller, right-click **User Programs** and click **Verify Programs**.
   Or, on the **Programs** menu, click **Verify**.
2. The results of the verify will appear in the **Verify Results** window. If errors are found, you must edit the program to fix the problem and verify again.
3. If no errors are found, download the User Programs to the controller.
4. After you have downloaded the User Programs, you may run them. See the Running User Programs topic for instructions.

4.1.4. Running User Programs

After you have created, verified and downloaded a User Program, you can run it on the RMC70.

Note:
In order to run User Programs or the PreScan Table, the RMC70 must be in RUN mode. See the RUN Mode and PROGRAM Mode topics for details.

Starting a User Program

There are 3 ways to start a User Program:

- **Issue the Start Task (90) command from RMCTools:**
    1. In the **Command Tool**, click the Axis 0 **Cmd** box. Type "90" and press enter.
    2. Enter the correct values in the **Task Number** and **Program** parameters. For details on these parameters, see the Start Task (90) command.
    3. Click **Send Command**.
- **Issue the Start Task (90) command from a User Program:**
    1. In the **Command** box, type "90" and press Enter.
    2. Enter the correct values in the **Task Number** and **Program** parameters. For details on these parameters, see the Start Task (90) command.
    3. Complete the step as described in the Creating User Programs topic.
- **Create a Scan Item in the PreScan Table**

How to Tell if a User Program is Running

Task Monitor

Use the Task Monitor to track which User Programs and steps the Tasks are currently running. See the Tasks topic for details.

To open the Task Monitor:

- In the Project pane, double click **Tasks**.

4.1.5. Labeling Steps

You can add a label to any step in a User Program. The label may be used in Link Types when specifying the step to jump to. Labels are global: you can jump to a label in any User Program. Labels make the User Programs more flexible and readable.

Creating and Changing Labels

The following procedures require that you first open a User Program in the Step Editor.

Add a Label to a Step

- Right-click anywhere in the step and click **Edit Label.**
- Type the name in the **Label** box and click **OK.**

Or,

- On the **Editor** menu, click **Edit Label**.
- Click **Add**.

Delete a Label

- On the **Editor** menu, click **Edit Label**.
- Click **Delete**.

Edit a Label

- Right-click anywhere in the step and click **Edit Label**.
- Make any changes and click **OK**.

Or,

- On the **Editor** menu, click **Edit Label**.
- Click the desired label and click **Edit**.
- Make any changes and click **OK**.

Using Labels to Jump to Steps

Jumping to a step by specifying a Label is often better than specifying a Step Number. If you use a step number, the number will change if a step is inserted before it. Then the jump will go to the wrong step. If you use a step Label, the jump will always go the step with that label.

To create a jump to a step with a label:

- Add a label to the step you wish to jump to.
- In the step you are jumping from, choose one of the **jump**Link Types.
- In the **Jump To**, **Jump on True**, or **Jump on False** box, tType the name of the label**.**

4.2. Expressions

4.2.1. Expressions Overview

Mathematical calculations can be performed on the RMC70 with *expressions*. Expressions provide for efficient calculations and compact User Programs. Use expressions in User Programs:

- In the Wait For and Advanced Condition Link Types.
- In the Expression (113) command.

Expression Types

There are two expression types. For details on each, click the respective type:

Type	Description	Used in Link Conditions	Used in Expression Command
Assignment Expression	Assigns a value to a register or variable.		✓
Logical Expression	Evaluates to True or False.	✓	

Parts of an Expression

The building blocks of all expression are:

Operators These are symbols that represent an action to be performed.

Functions These are predefined, named formulas.

Operands These are the elements that the operators and functions work on:

Type	Examples
Constant Value	3, 10.345
Register	F8:3, _Axis[0].ActPos
Variable	SampleVariable1 (must exist in the variable table)

Troubleshooting Expressions

For help on troubleshooting expressions, see the Troubleshooting Expressions topic.

Operators and Functions

For details on the operators and function used in expressions, see the Operators and Functions topics.

4.2.2. Assignment Expressions

An assignment expression assigns a value to a register. Assignment expressions are used only in the Expression (113) command in a User Program. For an overview of all expression types, see the Mathematical Expressions topic.

Expression Format

An assignment expression must follow this format:

Register := **Numeric Expression**

where

Register must be writable and may be any of the following:

- Variable
- Register - specified by its *tag name*
- Register - specified by an *address*

Numeric Expression is a mathematical expression that must evaluate to a numeric value. The syntax for a numeric expression is described below.

Sample Expressions

The following are valid expressions:

_VarTbl.CurVal[1]:=3.0 + Abs(_Axis[0].ActPos)

SampleVariable:=9.0 + Pos1/ (3.0 + _Axis[1].ActPF)
Note: SampleVariable must be defined in the Variable table.

See also:

Expressions Overview

Operators

Functions

4.2.3. Logical Expressions

A logical expression must evaluate to True or False. Logical expressions can only be used in Link Types.

Sample Expressions

The following are valid logical expressions:

_Axis[0].ActPos > Pos1 AND _Axis[1].ActPos < 4.0

SampleVariabel <> 20.0

See also:

Expressions Overview

Operators

Functions

Creating Expressions

Use the Expression Builder to create an expression.

To open the Expression Builder:

- Create or open a User Program.
- Go to the desired step.
- Choose the Expression (113) command in the **Command** box.
- Double-click the **Expression** box.

Building an Expression

After you have opened the expression builder, follow these steps:

- Enter an expression in the box. Select the Tags, Functions and operators from the respective boxes.
- To enter a numeric value, type it in. Note that you must include a decimal point if the data type is REAL. If the data type is DINT, the number cannot have decimal point.
- Data type matching is important. Data types do not mix, you must use the conversion functions.
- If the expression is valid, it will be black. If it is red, the expression is not valid.

See also:

Expressions Overview

Operators

Functions

4.2.4. Functions

This topic describes the functions available in the RMC70 expressions. These function can be used in both Assignment Expressions or Logical Expressions.

Function	Description	Input Parameter Data Types	Returns Data Type	Example
Min(*a*,*b*)	Returns the lesser of the values ***a*** and ***b***.	REAL or DINT. Both input parameters must be of the same type.	Same as the data type of the input parameters.	Min(3.0,5.0) returns 3.0 Min(2,-5) returns -5
Max(*a*,*b*)	Returns the lesser of the values ***a*** and ***b***.	REAL or DINT. Both input parameters must be of the same type.	Same as the data type of the input parameters.	Max(3,5) returns 5 Max(-7.0,5.1) returns 5.1
Abs(*a*)	Returns the	REAL or	Same as	Abs(3) returns 3

	absolute value of ***a***.	DINT	the data type of the input parameters.	Abs(-3.0) returns 3.0
Sqrt(*a*)	Returns the square root of ***a***.	REAL	REAL	Sqrt(4) returns 2
REAL_TO_DINT(*a*)	Rounds *a* to the nearest integer. 0.5 rounds away from zero.	REAL	DINT	REAL_TO_DINT(5.5) returns 6
DINT_TO_REAL(*a*)	Returns ***a*** in REAL type.	DINT	REAL	DINT_TO_REAL(5) returns 5.0
Trunc(*a*)	Rounds ***a*** towards 0.	REAL	DINT	Trunc(5.678) returns 5 Trunc(-5.678) returns -5

See also:

Operators

4.2.5. Operators

This topic describes the operators available in the RMC70 expressions. Each operator can be used in Assignment Expressions, Logical Expressions, or both.

For details on using the operators, see the Assignment Expressions and Logical Expressions topics.

Function	**Description**	**Assignment Expression**	**Logical Expression**
:=	Assigns the value of the expression on the right-hand side to the register on the left-hand side. Required in all assignment expressions.	✓	
+	Addition	✓	✓
-	Subtraction	✓	✓
*****	Multiplication	✓	✓
/	Division	✓	✓
(	Opening parenthesis. The number of opening parentheses must match the number of closing parentheses.	✓	✓
)	Closing parenthesis. The number of opening parentheses must match the number of closing parentheses.	✓	✓
=	Equal To (not for assigning values)		✓
<>	Not Equal		✓
<=	Less Than or Equal To		✓

<	Less Than		✓
>	Greater Than		✓
>=	Greater Than or Equal To		✓
AND	Logical AND	✓	✓
OR	Logical OR	✓	✓
NOT	Logical NOT	✓	✓
XOR	Logical Exclusive OR	✓	✓

See also:

Functions

4.2.6. Troubleshooting Expressions

The Expression Builder displays the Assignment Expression in black text if it is invalid, and in red text when it is invalid. Listed below are common errors in the assignment expression syntax:

Mixed Data Types:

Data types cannot be mixed in the assignment expression without explicitly converting them. A number that *does not* contain a decimal point is considered to be a DINT data type. A number that *does* contain a decimal point is considered to be a REAL data type.

Example1:

Assume **MyVariable** is defined to be a REAL type. The following expression *is not* valid, because "3" is a DINT type:

```
MyVariable:=_Axis[0].ActPos+3
```

The following expression *is* valid because "3.0" is a REAL type:

```
MyVariable:=_Axis[0].ActPos+3.0
```

The following expression *is* valid because the DINT_TO_REAL() function converts "3" to a REAL type:

```
MyVariable:=_Axis[0].ActPos+DINT_TO_REAL(3)
```

Example2:

Assume **MyVariable** is defined to be a DINT type. The REAL_TO DINT() function converts the REAL type part of the expression to a DINT. The following expression *is not* valid because "2" is a DINT type and _Axis[0].ActPos is a REAL type:

```
MyVariable:=REAL_TO_DINT(Abs(_Axis[0].ActPos/2))
```

The following expression *is* valid because "2.0" is a REAL type and _Axis[0].ActPos is also:

```
MyVariable:=REAL_TO_DINT(Abs(_Axis[0].ActPos/2.0))
```

Similarly, the following expression is *not* valid because the Abs() function returns the data type of its arguments, which in this case is REAL, while MyVariable is a DINT type:

```
MyVariable:=Abs(_Axis[0].ActPos/2.0)
```

Illegal Expression Order

An expression *must* begin with "[register]:= ", where register is any register in the RMC70 and may be given as a register address, tag name, or variable name. See the Assignment Expression topic for details.

4.3. Link Types

4.3.1. Link Types Overview

A Link Type specifies the condition that makes the RMC jump to and start the next step in a User Program. As a User Program runs, the RMC checks the Link Type of the current step every control-loop time. When the Link Type evaluates to true, it jumps to the specified step. The step it jumps to is specified by a valid step number or a step label.

For example, you can set a link type to wait until the axis is in position before going to the next step, or you may want to wait a certain amount of time. Complex link conditions can also be specified.

There are two basic categories of Link Types:

- Link Types that jump to the *next* step.
- Link Types that jump to *any* step. These Link Types have the word "jump" in their name.

Selecting a Link Type

To select a link type, go to the desired step in a User Program and double-click the **Link Type** box. Choose one of the link types.

List of Link Types

Link Type	Description
End	Ends the sequence. No steps are executed after this step.
Immediate	Jumps immediately to the next step. **Note:** The next step is executed in the next loop-time of the RMC.
Delay	Waits for number of seconds specified in the **Link Condition** box and then jumps to the next step. The **Link Condition** box must evaluate to a numeric value. It can be any of the following: • Constant value • Register • Variable • Numeric Expression
Wait For	Waits for the **Link Condition** to become true and then jumps to the next step. The **Link Condition** box must be a logical expression that evaluates to True or False.
Jump	Jump immediately to the step specified in the **Jump to** box. The **Jump to** box must contain a valid step number or label.
Delay Jump	Waits for number of seconds specified in the **Link Condition** box and then jumps to the step specified in the **Jump to** box.

Conditional Jump	Evaluates the **Link Condition**. If it is True, the **Jump On True** is taken. If it is false, the **Jump On False** is taken. The **Jump On True** and **Jump On False** boxes must contain a valid step number, label, or "Wait". The **Link Condition** box must be a logical expression that evaluates to True or False.

4.3.2. Link Type: End

> **Note:**
> A Link Type specifies the condition that makes the RMC jump to and execute the next step in a User Program.

The End Link Type ends the sequence. No steps are executed after this step.

To select the End Link Type:

- Open or create a User Program.
- Go to the step where you want the End Link Type.
- Double-click the **Link Type** box.
- Click **End**.

4.3.3. Link Type: Immediate

> **Note:**
> A Link Type specifies the condition that makes the RMC jump to and execute the next step in a User Program.

The Immediate Link Type immediately jumps to the next step in the User Program. The next step is then executed in the next loop-time of the RMC.

To select the Immediate Link Type:

- Open or create a User Program.
- Go to the step where you want the Immediate Link Type.
- Double-click the **Link Type** box.
- Click **Immed**.

4.3.4. Link Type: Delay

> **Note:**
> A Link Type specifies the condition that makes the RMC jump to and execute the next step in a User Program.

The Delay Link Type waits the number of seconds specified in the **Link Condition** box before jumping to the next step in the User Program.

Boxes

The Delay Link Type has the following boxes:

Box	Description
Time to Delay	The Time to Delay specifies the number of seconds to wait before jumping to the next step in the User Program. It must evaluate to a numeric value and may be any of the following: • Constant value • Register • Variable • Numeric Expression

To select the Delay Link Type:

- Open or create a User Program.
- Go to the step where you want the Delay Link Type.
- Double-click the **Link Type** box.
- Click **Delay**.

4.3.5. Link Type: Wait For

> **Note:**
> A Link Type specifies the condition that makes the RMC jump to and execute the next step in a User Program.

The Wait For Link Type waits for the **Link Condition** to become true before jumping to the next step in the User Program.

Boxes

The Wait For Link Type has the following boxes:

Box	Description
Link Condition	The Link Condition specifies the condition that must become true before jumping to the next step in the User Program. This condition must be a logical expression that evaluates to True or False.

To select the Wait For Link Type:

- Open or create a User Program.
- Go to the step where you want the Wait For Link Type.
- Double-click the **Link Type** box.
- Click **Wait For**.

4.3.6. Link Type: Jump

> **Note:**
> A Link Type specifies the condition that makes the RMC jump to and execute the next step in a User Program.

The Jump Link Type jumps to the step specified in the **Jump On True** box. The next step is then executed in the next loop-time of the RMC.

Boxes

The Conditional Link Type has the following boxes:

Box	Description
Jump to	Specifies the step to jump to. It must be a valid step number or step label.

To select the Advanced Condition Link Type:

- Open or create a User Program.
- Go to the step where you want the Advanced Condition Link Type.
- Double-click the **Link Type** box.
- Click **Jump**.

4.3.7. Link Type: Delay Jump

> **Note:**
> A Link Type specifies the condition that makes the RMC jump to and execute the next step in a User Program.

The Delay Link Type waits the number of seconds specified in the **Time to Delay** box before jumping to the step specified in the **Jump** box.

Boxes

The Conditional Link Type has the following boxes:

Box	Description
Time to Delay	The Time to Delay specifies the number of seconds to wait before jumping to the next step in the User Program. It must evaluate to a numeric value and may be any of the following: • Constant value • Register • Variable • Numeric Expression
Jump	Specifies the step to jump to. It must be a valid step number or step label.

To select the Delay Jump Link Type:

- Open or create a User Program.
- Go to the step where you want the Delay Link Type.
- Double-click the **Link Type** box.
- Click **Delay**.

4.3.8. Link Type: Conditional Jump

> **Note:**
> A Link Type specifies the condition that makes the RMC jump to and execute the next step in a User Program.

The Conditional Jump Link Type evaluates the **Link Condition**. If it is True, the program jumps to the step specified in the **Jump On True** box. If it is false, the program jumps to the step specified in the **Jump On False** box.

Boxes

The Conditional Link Type has the following boxes:

Box	Description
Link Condition	The Link Condition specifies the expression that is evaluated. It must be a logical expression that evaluates to True or False.
Jump On True	Specifies the step to jump to if the Link Condition is True. It must be a valid step number, step label, or "Wait".
Jump On False	Specifies the step to jump to if the Link Condition is False. It must be a valid step number, step label, or "Wait".

To select the Advanced Condition Link Type:

- Open or create a User Program.
- Go to the step where you want the Advanced Condition Link Type.
- Double-click the **Link Type** box.
- Click **Adv Cond**.

4.4. Programming Overview

The RMC70 has a rich set of pre-programmed high-level motion commands. In addition, it can easily be programmed to perform simple motions or complex actions. With the RMC70 features listed below, simple applications can be done entirely within the RMC70, or in conjunction with a PLC.

> **Note:**
> In order to run User Programs or the PreScan Table, the RMC70 must be in RUN mode. See the RUN Mode and PROGRAM Mode topics for details.

User Programs

A User Program carries out a sequence of commands without requiring a PLC or other controller. This allows the RMC to respond to events within its control-loop time rather than the scan rate of the PLC. It also reduces the controller programming required.

A User Program consists of multiple steps. Each step can issue a command on one or several axes. The series of steps are linked together in sequences. The link types allow branching and looping, waiting for conditions and many other features. Math operations are also possible in the user program. An RMC70 controller may execute several User Programs simultaneously.

For details on creating and running User Programs, see the User Programs topic.

PreScan Table

The PreScan Table is a mechanism for setting up conditions that start or stop Tasks. The conditions are checked every scan (loop time). When a condition becomes true, the User Program(s) specified by the user are started.

For example, a condition can check if the Fault input is on. When it comes on, it can start an error handling Task, and stop all the other Tasks. The PreScan table can also handle complex conditions. For example, a condition can check for certain inputs to be on *and* the Actual position to be greater than a certain value.

See the PreScan Table topic for details.

Variables

Variables make the User Programs very flexible. Variables can be used to effortlessly change programs, make programs readable, and easily influence User Programs via a PLC. Variables may be used in command parameters, the Expression (113) command, and several Link Types.

See the Variables topic for details.

Commands

The commands are the building blocks of RMC70 programming. Commands tell the RMC70 what to do. For a list of commands, see the Command List topic. Commands may be issued from the following places:

- From RMC70Tools using the Command Tool.
- From a PLC or other host controller via the communication port.
- From a User Program.

Programming from External Systems

The RMC70 can be controlled from a PLC or other host controller. It can be done in the following ways:

Entirely with a PLC

A PLC can exercise complete control over the RMC70 by issuing commands to it. The RMC70 supports many communication protocols, such as DF1, Modus/RTU and PROFIBUS.

With the RMC and PLC (or HMI)

The RMC70 can be programmed using a combination of a host PLC (or HMI) and User Programs. This allows fast and time-critical sequences to be implemented in the User Program (which executes at the RMC70 loop time), while less time critical functions are handled via the PLC.

For details on using the RMC70 with an HMI, see the Communicating with HMIs topic.

4.5. Tasks

A Task is defined as an execution engine. Each Task, or engine, of the RMC70 is capable of running one step in a User Program at any given moment. The RMC70 has up to four Tasks. Therefore, the RMC70 can run up to four User Programs simultaneously.

> **Note:**
> In order to run Tasks or the PreScan Table, the RMC70 must be in RUN mode. See the RUN Mode and PROGRAM Mode topics for details.

Why Tasks?

If multiple User Programs are running simultaneously, some method of keeping track of them is required, even when a step in one User Program jumps to another User Program or even starts another User Program. If you want to stop a User Program, you need some way to stop it even if it's jumping from one User Program to another. The Tasks help keep track of all of this.

Starting Tasks

Tasks can be started two different ways:

- Use the Start Task (90) command to start a Task. The Start Task command starts the specified Task at a User Program. The following items are important when issuing a Start Task command:
  - Which axis to issue the Start Task command to

    In each step of the User Program you will run, if you have not specified which axis the command will be issued to, then it will be issued to the axis on which the Start Task command was issued. If you *have* specified for each step which axis the command will be issued to, then it does not matter which axis you issue the Start Task command to.
  - Selecting which Task to run

    When you issue the Start Task command, you must specify which Task to start. What happens depends on what the Task is doing before you issue the Start Task command:
    - **The Task is stopped.**
      If you select a Task that is stopped, the Task will simply start running at the label you specified.
    - **The Task is running.**
      If you select a Task that is running, it will stop and immediately start at the program you specified.
  - Select the User Program

    The second command Parameter, **Program**, specifies which User Program the Task will start at.
- A Scan Item in the PreScan Table can start a Task. See the PreScan Table topic for details.

Stopping Tasks

Tasks can be started two different ways:

- Use the Stop Task (91) command. The Stop Task command immediately stops the specified task. The User Program currently running on the Task will immediately stop.
- A Scan Item in the PreScan Table can start a Task. See the PreScan Table topic for details.

Task Monitor

Use the Program Monitor to see which Tasks are currently running User Programs. The Program Monitor shows each Task, its state, and which Program and step it is currently on or the last step it was on. The Task Monitor also display the current values of the variables in the variable table. See the Task Monitor topic for details.

Changing the Number of Tasks

By default, the RMC70 has 2 Tasks. To increase the number of Tasks:

- In the Project pane, right-click **Tasks** and click **Properties**.
- Select a number in the **Number of User Tasks** box and click **OK**.
- If you changed the number of Tasks, the controller must be restarted for the changes to take affect. Click **Yes** to restart the RMC70.

4.6. Variables

The RMC70 has 64 registers that are specifically intended for user-defined variables. Variables may be named by the user and can be set to any value at any time. Variables make the User Programs very flexible. Variables can be used to effortlessly change programs, make programs

readable, and easily influence User Programs via a PLC. Variables may be used in command parameters, the Expression (113) command, and several Link Types.

Example:

If you create a variable called Speed, and enter it as the Speed command parameter in many Move commands in a User Program, then you can easily change the speed for all those commands at once by simply changing the value of the Speed variable.

Example:

To count how many times an event occurs, you can add 1 to a variable each time the event occurs.

Features

Name You can name a variable. Use the Variable name to refer to the variable. You can also reference the variable by its register address, especially if communicating to the RMC70 with a PLC.

Current Value This is the value of the variable. There are several ways to change the current value:

- Use the Variable Table Editor
- Use the Expression (113) command
- Use the Write Register (112) command
- Use a PLC or other host controller

Initial Value You can specify an initial value for a variable. The variable is set to its default value when the RMC is powered up.

Creating Variables

To create variables and set the initial value, use the Variable Table Editor.

Assign a Value to a Variable

There are several ways to assign a value to a variable:

- Use the Variable Table Editor to set the initial value. The Initial value is the value the variable will take on when the RMC70 is reset.
- Use the Program Monitor to change the current value of the variable.
- Use the Expression (113) command in a User Program to assign a value to a variable.
- Write to the Variable register via a PLC or other host controller. See **Using Variables with a PLC or other Host Controller** below for details.

Using Variables

Variables can be used for many purposes:

- Store a value for later use.
- To influence User Programs.
- Count how many times an event occurs.

To use the values assigned to variables, do the following:

- Enter variables in the Command parameters in User Programs.
- Use variables in Link Type expressions in Use Programs.
- Use variables in the Expression (113) command in User Programs.
- Use variables in the PreScan Table condition.

Using Variables with a PLC or other Host Controller

You can read or write to variables when communicating to the RMC70 with a PLC or other host controller. The register addresses of the variable registers are F56:0 to F56:63. Use Variable Table Editor to find the address for each variable. Also, see the Register Map for address details.

4.7. PreScan Table

The PreScan Table is a mechanism for setting up conditions that start or stop Tasks. The conditions are checked every scan (loop time). When a condition becomes true, then a Task Action - as set up by the user - is started.

For example, a Scan Item may be set up to check if the Fault input and a Digital I/O input is on. When they both come on, it stops the current Task that is running and starts a Task for handling the error condition.

If the PreScan Table is enabled, the PreScan Items are automatically checked every scan. This makes the PreScan Table very useful for error handling, such as starting an error routine when an Auto Stop turns on.

Scan Items

The PreScan Table can contain multiple **Scan Items**. A Scan Item is a complete entry in the table, consisting of a **condition** and a **Task Action**:

Conditions

The PreScan table can handle simple or complex Conditions. For example, a Condition can check for certain inputs to be on *and* the Actual position to be greater than a certain value. Conditions are created using expressions and are therefore very flexible. For details on expression, see the Expressions topic. To find out how to create a Conditon, see **Creating a Scan Item** below.

Task Actions

Task Actions can start or stop Tasks. Tasks run User Programs. For each Scan Item, you can define which Tasks to start, stop, or do nothing to. The PreScan Table will allow a Task to be started only at a label in a User Program. To find out how to set up the Task Action, see **Creating a Scan Item** below.

Creating a Scan Item

To create a Scan Item:

- In the Project Pane, expand **Programming**, and double-click **PreScan Table**.
- In the **Condition** column, double-click the cell with an asterisk next to it. The New Condition Wizard will open.
- Select the type of condition you want to create, and continue through the wizard to complete the condition.

> **Note:**
> If you want to create a complex condition, choose **Other**. It will let you create a custom condition. See the Expressions topic for details on creating expressions.

- In the PreScan Table, in each **Task** cell for the condition you just entered, click the cell and select one of the following options from the drop-down box:
  - **STOP**
    The Task will be stopped immediately when the condition becomes true.
  - **A Label**
    Choose a label from the drop-down box. In order top use labels, you must first have created a label in a User Program.

- **No Entry**
  If you do not want the Scan Item to affect the Task, do not enter anything in the cell. To remove an entry, click the cell and press Delete.

- In the Project Pane, right-click **Programming**.
- Select **Enable the PreScan Task** and click OK.
- Make sure you are online with the controller.
- In the Project Pane, right-click **Programming** and click **Download Programs**.
- You are now finished. When the condition becomes true, the Task Action will be started.

4.8. Data Types

The RMC70 data have the following internal types:

Data Type	Description
REAL	32-bit floating point number
DINT	32-bit Integer number
DWORD	32-bit string of bits

When used in expressions, Data Types can not be mixed without explicitly converting them. To find the data type of a register, use the Address Selection Tool.

Accessing Data Types Externally

When reading and writing registers in MC70 via one of the communication protocols, use the data type as defined by the communication protocol, not the internal data type of the RMC70.

Serial DF1:

The RMC70 registers are located in the F file, which is 32-bit floating point numbers. The RMC70 registers all appear as floating point to the DF1 protocol, even though they may be defined as a DINT or DWORD internally to the RMC70.

5. Using RMC70Tools

5.1. Using RMC70Tools

RMC70Tools is a Windows 98/NT/2000/XP based software package integral to the RMC70 series motion controllers. It allows the user to set up, display, troubleshoot, program and control all features of the RMC70 motion controller. Fully detailed plots of motion can be captured at any time. RMC70Tools offers high speed communications to the RMC via a serial cable, allowing the user to tune even the most time-critical applications.

RMC70Tools is an Integrated Development Environment. All the editors and tools are part of the environment and the data is saved as part of the project. The plots in the Plot Manager may be saved individually.

Requirements

RMC70Tools requires Windows 98, Windows NT 4.0, Windows 2000, or Windows XP.

RMC70Tools components

RMC70Tools IDE

Menu

Project Pane

Output Pane

Axis Status Monitor

Axis Parameter Editor

Custom Data Map Editor

Step Editor

Variable Table Editor

Indirect Data Map Editor

Event Log Monitor

Command History

Tools

Address Tool

Command Tool

Plot Manager

Wizards

New Controller

Actions

Starting a project

Connecting RMC70Tools to an RMC

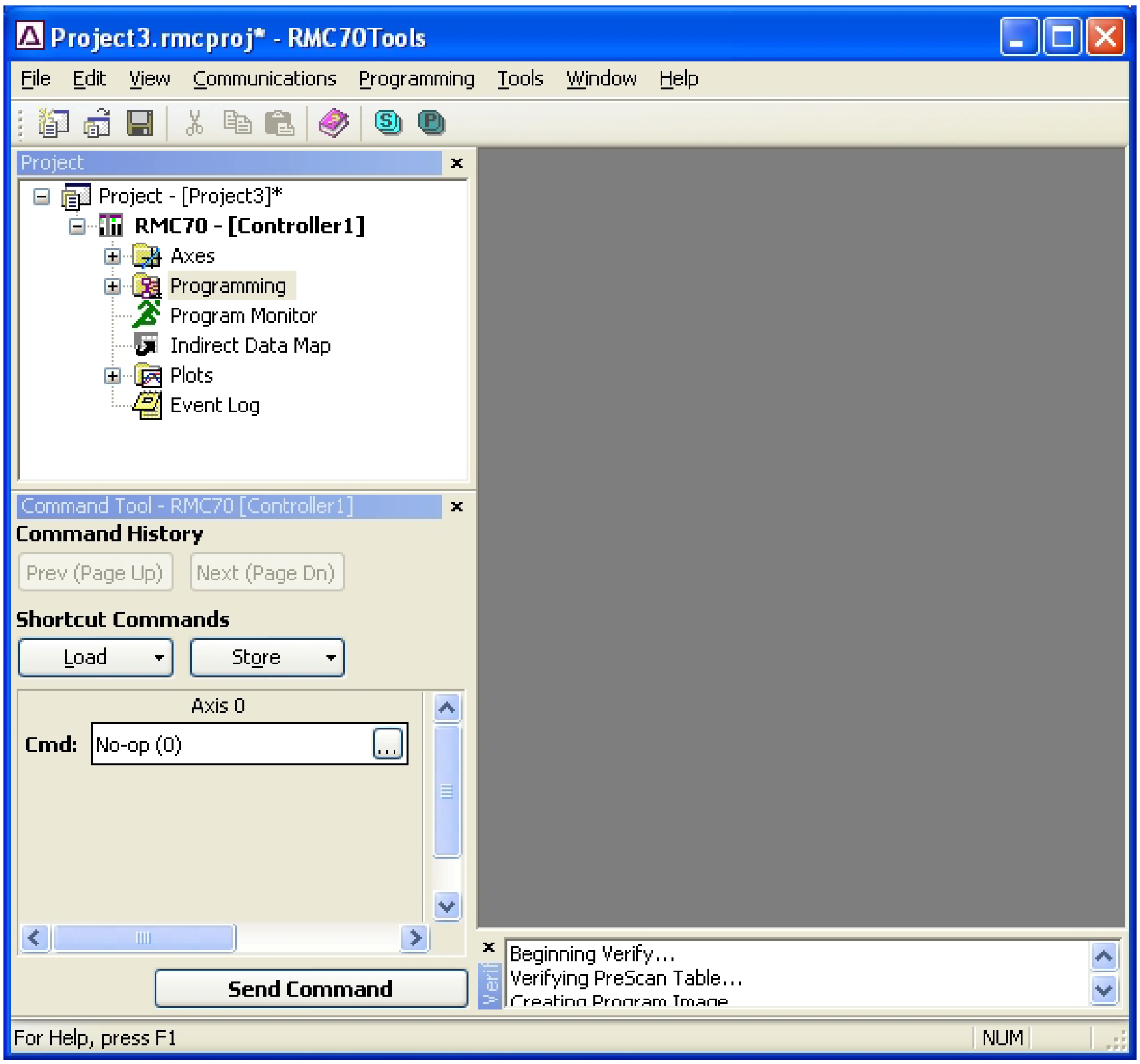
Adding a Motion Controller

5.2. RMC70Tools IDE

Many dockable panes and non-dockable windows may be open in the RMC70Tools integrated development environment. The sample below is a recommended basic layout. The dockable Project pane should always be open as it provides an immediate overview of the project and all components can be accessed from it. Placing the Command Tool below it provides an easily accessible method of issuing commands.

The open space to the right can be used for other windows, such as the Axes Parameter Editor and Axes Status Monitor which display important registers in the controller.

The Program folder includes the Step Editor, Variable Table and PreScan Table Editor, all of which can be used to create and edit User Programs. The Event Log is useful for troubleshooting.



5.3. RMC70Tools Menu Bar

This topic covers only the non-intuitive items in the RMC70Tools menu bar. The RMC70Tools menu bar consists of the following menus:

Components

File

New

Controller Opens the New Controller Wizard which adds a new controller to the project. The controller must first be connected to the computer's serial port via a cable.

Project Opens a new project. You will be prompted to first save the current project.

Edit

View

Command History

Opens the Command history window.

Properties

Opens the Properties dialog for the currently selected item in the Project pane.

Controller

Go Online

Starts communication with the currently selected controller. If the communication is broken, RMC70Tools will not attempt to restart communication.

Go Offline

Closes communication with the currently selected controller.

Upload All

Uploads all the parameters from the currently selected controller to the project. This will overwrite the project values.

Download All

Downloads all the parameters from the project to the currently selected controller. This will overwrite the controller values.

Standby Mode

In this mode, if communication is broken, RMC70Tools will automatically attempt to restart communication until it is successful. This mode is not yet available.

Tools

Window

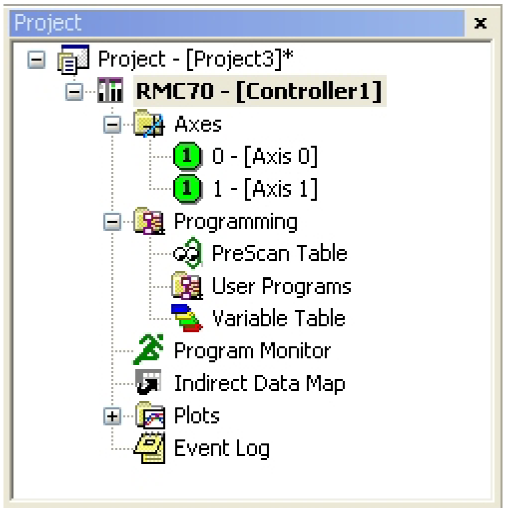
Help

5.4. RMC70Tools Project Pane

To access this pane:
The Project pane should already be open in RMC70Tools. If it is not, on the View menu, click **Project**, or press Alt+0.

The Project pane provides an hierarchical overview of the components of the project. All components can be accessed from the Project pane by double-clicking or right-clicking the item. For ease of use, leave this pane open when using RMC70Tools.

The Project pane shows all of the controllers in the project and their components. The example below contains one controller:



Components

Each RMC70 controller has the following components, visible in the Project Pane:

Axes

Shows the axes of the controller. Right-click an Axis to edit the axis' name.

Double-click any axis to open the Axes Parameter Editor and Axes Status Monitor

Programming

Contains items for programming the controller.

Double-click PreScanTable to open it.

Multiple User Program may be created for each controller. Double click User Programs to create a new program.

Double-click Variable Table to open it.

Program Monitor

Double-click Program Monitor to open it.

Indirect Data Map

Double-click the Indirect Data Map to edit it.

Plots

Double-click a plot to open it. See Plot Manager for details on using the plots.

Event Log

Double-click to open the Event Log. This log is useful for troubleshooting. It logs issued commands, changed parameters, and errors.

5.5. Axis Tools Pane

To access this pane:
On the RMC70Tools toolbar, click the **Axis Tools** button ( ).

The Axis Tools pane displays several types of information for the RMC70. It consists of two panes:

- **Axis Status Registers**
  The Status Registers provide information on the status of each axis in the RMC70. For information on each register, click on the register cell in any **Axis** column, and press **F1**.
  For details on the Axis Status Registers pane, see the Axes Status Registers Pane topic. For a list of all the Status Registers, see the Register Map topic.
- **Axis Parameters**
  The Axis Parameters provide configuration information for the RMC70. For information on each parameter, click on the parameter cell in any **Axis** column, and press **F1**.
  For details on the Axis Parameters pane, see the Axes Parameters Pane topic. For a list of all the Axis Parameter registers, see the Register Map topic.

Using the Axis Tools Pane

Auto Refreshing

When the **Auto Refresh** check box is selected, the Axis Tools pane constantly updates its data when RMC70Tools is online with the RMC70. To disable Auto Refresh, clear the **Auto Refresh** check box. Disabling Auto Refresh may conserve communications resources.

Choosing Views

Use the **Select Layout** box ( ) to choose the pane layout in the Axis Tools. The various layouts may help you place the Axis Tools pane such that other panes in RMC70Tools are also visible.

Hiding and Showing Columns

Use the **Select Axes** box ( ) to choose which axes to display.

Showing Project or Controller Values

To view the values in the RMC70Tools project, click the **Show Project Values** button ( ). To view the values in the RMC70, click the **Show Controller Values** button ( ). If the values in the RMC70 and in the RMC70Tools project are identical, the **Show Controller Values** button will be disabled.

Uploading and Downloading Values

After making changes to any editable parameters in the Axis Tools, you must download the changes to RMC70. To download the project values from the Axis Tools to the RMC70, click the Download button ( ). To upload the controller values from the RMC70 to the project, click the Upload button ( ).

5.6. Event Log Monitor

To access this dialog:
Double-click the desired controller in the Project Pane and Double-click **Event Log**.
Or:
Press Alt+0 to open the Project Pane. Use the arrow keys to select and expand the desired controller and then highlight **Event Log**. Press Enter.

Tip:
The Event Log monitor is very useful for troubleshooting. Use it to find out if commands were issued correctly, if and when errors occurred, and when parameters changed.

The Event Log Monitor displays all events that have occurred in the controller, such as issued commands, changed parameters and errors. The Event Log Monitor is an important aid in troubleshooting.

Using the Event Log

The Event Log helps you determine what happened in the controller. Here are some things it can be used for:

- See if a command was successfully issued. The entire command, with parameters, is displayed.
- Find out which, if any, error occurred.
- See where a command was issued from, for example, from a PLC, from a User Program, from the Command Tool.

Entry Details

Each entry in the Event Log Monitor has the following components:

Component	Description
#	The number is solely for indicating the order of the events.
Time	The time the event occurred is given in the following format: *day*d *hour*:*minute*:*second*:*millisecond* The time is measured from when the RMC was last powered up.
Event	Describes the type of event.
Details	Provides additional details on the event, such as the previous and changed values, command source, etc

5.7. Uploading and Downloading Axes

To upload or download data to or from the controller:

- Right-click the desired folder or item in the Project pane. Click **Upload *item*** or **Download *item***.

Or,

- In some editors, click the Download or Upload button in the toolbar.

Downloading

Downloading applies the data in the Project file to the RMC. The data in the RMC will be overwritten.

Uploading

Uploading applies the controller data in the RMC to the Project file in RMC70Tools. The data in the Project file will be overwritten.

Why Bother?

A Project file can store all the information in an RMC. This allows a copy of the controller data to be stored externally. When changing the controller parameters, a backup of the original parameters is still available in the Project file. Uploading and downloading transfers the data from the RMC to RMC70Tools, or vice versa.

5.8. Command Tool

> **To access this tool:**
> If the Command Tool is not already open, click **Address Tool** on the **Tools** menu, or press Alt+8.
> The Command Tool is dockable and can be resized and moved as desired.

Issuing Commands

Issue a command using the following steps:

a. Select a command for the desired axis by either typing it in the **Cmd** box and selecting from the list that opens upon typing, or by clicking the ellipsis button ... and choosing from the command tree.
b. Enter values in the parameters. Each command has its own parameters. Some commands may not have any parameters.
c. Repeat a and b for the other axis. If no command is desired, choose No-op(0).
d. Click **Send Command** to issue commands to the axes. It is not possible to issue a command to only one axis.

Using Shortcut Commands

Shortcut Commands can be used to quickly issue a command to an axis. This is useful when setting up and tuning an axis, which usually requires moving repeatedly between two or more positions.

Issuing Shortcut Commands

Shortcut Commands can be issued in two ways:

- Hold down CTRL and press the number of the stored command you wish to execute: 0 to 9.
- Click **Load**, choose the shortcut command, and click Send Command.

Setting up Shortcut Commands

To set up a Shortcut Command, choose a command and enter values for the command parameters for each axis. Click **Store** and choose the Shortcut Command you would like to store it as, for example, CTRL+6.

> **Note:**
> The commands on both axis will be stored. If no command is desired on the other axes, select the No-op(0) command.

Viewing Shortcut Commands

To view the Shortcut Commnads, click **Load** or **Store** and click **Display...**.

5.9. Output Pane

> **To access this pane:**
> On the View menu, click **Output**, or press Alt+1.

The Output pane is a dockable pane in RMC70Tools. The output pane has the following functions:

- Displays the results of downloading all settings to the RMC70.

5.10. Axes Status Registers Pane

To access this pane:
On the RMC70Tools toolbar, click the **Axis Tools** button ( ). If the **Axes Status Registers** pane does not appear, use the **Select Layout** box ( ) to choose the **Axes Status Registers** pane.

The **Axis Status Registers** pane is part of the Axis Tools pane. It displays the Axis Status registers. This data is very useful in all phases of using the controller.

Tabs

The Axes Parameters pane has 3 tabs:

- **Basic Position**
  Use this tab to view the basic status registers for a position axis.
- **Basic Velocity**
  Use this tab to view the basic status registers for a velocity axis.
- **All Tab**
  Use this tab to view all the status registers. This tab is useful for advanced troubleshooting.

Within each tab, the registers are arranged in sections. To expand a section, clicking the '+'.

Using the Axes Status Registers Pane

Columns

Column Name	Description
Registers	This is the descriptive name of the parameter register.
Reg #	The Reg# column contains the address number for each Axis 0 register. For details on register addresses, see the Register Address Format topic. **Note:** The address **F12** refers to Axis 0. The register addresses for Axis 1 are **F13**. Otherwise, they are identical to Axis 0.
Units	These are the units of the register in the **Reg #** column.
Axis	This is the current value for this axis of the register in the **Reg #** column.

5.10.1.1. Hiding and Showing Columns

To hide a column, right-click the column heading and click **Hide Column**. To see it again, right-click any column heading, click **Add Column**, and click the desired column name.

5.11. Step Editor

The Step Editor

5.12. Program Monitor

To access this monitor:
Expand the desired controller in the Project pane, right-click **Programming**, and right-click **Program Monitor**.

Use the Program Monitor to view the current state of the User Programs. You can view the current step for each Task, and view and edit the current value of each Variable tag.

The Program Monitor is divided into two panes: the **Task** pane and the **Variables** pane.

Task Pane

Use the Task pane to view the current state of the User Tasks.

Columns

Column	Description
Task	This is the Task number.
State	Tells whether the Task is running or stopped.
Step	This is the current User Program and Step the Task is at. It is displayed as ***p:s*,** where ***p*** = Program number, and ***s*** = Step number.
Def. Axis	Default Axis. This is the axis that the Start Task command that started this Task was issued to. If a command in a User Program does not have any selected Commanded Axes, the command will be issued to this Default Axis.

Variables Pane

Use this pane to view and edit the current value of the variables.

Editing the Current Value of a variable

- Make sure you are online with the controller.
- In the Current column, enter a number.
- Click the Download button.

Monitor Axes Check box

Select this box to continually monitor the state of the axis. If this check box is cleared, the data in the window may not reflect the current state of the axis.

5.13. Shortcut Keys

Use the shortcut keys listed in this topic to navigate RMC70Tools without a mouse.

General

Key	Action
F1	Help
Tab	Next Item
Shift Tab	Previous Item

Edit

Key	Action
Ctrl + C	Copy
Ctrl + V	Paste
Ctrl + X	Cut

File

Key	Action
Ctrl + N	Create a New Project
Ctrl + O	Open a Project
Ctrl + S	Save the Project

Commands (Global)

Key	Action
Ctrl + Shift + I	Initialize all Axes
Ctrl + K	Disable All Axes
Ctrl + Shift + C	Clear All Axis Faults
Ctrl + number	Issue a Stored Command (number = 0-9)

Communications

Key	Action
Ctrl + Shift + O	Go Online

5.13.1.1. Programming

Key	Action
F7	Verify Programs

View and Tools

Key	Action
Alt + 0	Open Project pane
Alt + 1	Open Output pane
Alt + 2	Open Verify Results pane
Alt + 8	Open Command Tool
Ctrl + H	Open Command History
Ctrl + T	Open Axes Tools

Step Editor

Key	Action
Ins	Add Step Before
Alt + Ins	Add Step After
Ctrl + Del	Delete Step
Ctrl + Q	Edit Comment

Ctrl + L	Edit Label

Window Control

Key	Action
Ctrl + F6 or Ctrl + Tab	Next Window
Ctrl + Shift + F6 or Ctrl + Shift + Tab	Previous Window
Ctrl + L	Edit Command
Ctrl + Q	Edit Label
Ctrl + F4	Close Window
Alt + F4	Close Application

Project Pane

Key	Action
Alt + 0	Open Project pane
Right Arrow or Keypad +	Expand an Item
Left Arrow or Keypad -	Collapse an item
Enter	Open an item
Shift + Enter	Properties
Shift + F10	Shortcut menu

Plot Manager

Key	Action
Ctrl + N	New Plot Archive
Ctrl + O	Open Plot Archive
Ctrl + S	Save Plot Archive
Ctrl +Del	Delete Plot
In plot pane:	
Arrow Keys	Move cursor
Page Up	Move cursor to left
Page Down	Move cursor to right
Home	Move cursor to end of plot
End	Move cursor to beginning of plot
Alt +	Close Program Monitor

F4

Axis Tools and Indirect Data Map

Key	Action
Ctrl + D	Download Parameters
Ctrl + U	Upload Parameters

Command Tool

Key	Action
Ctrl + Shift + number	Store current command (number = 0-9)

Program Monitor

Key	Action
Ctrl + D	Apply Variable Changes

5.14. Go Online, GO Offline

The **Go Online** command in RMC70Tools connects RMC70Tools to an RMC. The **Go Online** stops communication between RMC70Tools and the RMC.

Connecting to a Controller

- First, connect a null-modem cable from the computer to the RS-232 monitor port on the RMC. See the RS-232 Monitor Port for details.
- To connect to a controller from RMC70Tools, right-click the desired controller in the Project pane and choose **Go Online**. RMC70Tools will attempt to connect to the controller via the controller's RS-232 monitor port. If the connection is successful, RMC70Tools checks to verify that the project settings are the same as the controller settings. If they are not, the Hardware Conflict dialog opens. This dialog alerts the user to whether something has changed in the project or controller, or perhaps the wrong controller is connected.
- To disconnect a controller from RMC70Tools, right-click the desired controller in the Project pane and choose **Go Offline**. RMC70Tools will cease communicating with the RMC.

How to Determine Whether a Controller is Online

The controller icon in the Project pane indicates whether RMC70Tools is communicating with a controller.

The controller icon looks like this when it is online (not communicating with RMC70Tools).

The controller icon looks like this when it is offline (communicating with RMC70Tools).

5.15. Editors

5.15.1. Axes Parameters Pane

To access this pane:
On the RMC70Tools toolbar, click the **Axis Tools** button ( ). If the **Axes Parameters** pane does not appear, use the **Select Layout** box ( ) to choose the **Axes Parameters** pane.

The **Axis Parameters** pane is part of the Axis Tools pane. Use the **Axes Parameters** pane for the following:

- Monitor and edit the Axis Parameters.
- Set up and tune the RMC70.

For general information on using editors, see the Using Editors topic.

Tabs

The Axes Parameters pane has 3 tabs:

- **Setup Tab**
  Use this tab for setting up the controller. This tab contains the basic parameters required for setup.
- **Tuning Tab**
  Use this tab for tuning the controller. This tab contains the basic parameters required for tuning.
- **All Tab**
  Use this tab to edit advanced parameters. This tab contains all the Axis parameters.

Within each tab, the registers are arranged in sections. To expand a section, clicking the '+'.

Using the Axes Parameter Pane

Editing Parameter Values

- Find the parameter you wish to edit.
- Click the cell in the desired **Axis** column.
- Type or choose a value and press Enter.
- Click the **Download** button to apply the changes to the RMC70.
- To save the values to FLASH memory, on the **Controller** menu, click **Update FLASH**.

Columns

Column Name	Description
Registers	This is the descriptive name of the parameter register.
Reg #	The Reg# column contains the address number for each Axis 0 register. For details on register addresses, see the Register Address Format topic. **Note:** The address **F12** refers to Axis 0. The register addresses for Axis 1 are **F13**. Otherwise, they are identical to Axis 0.
Units	These are the units of the register in the **Reg #** column.
Axis	This is the current value for this axis of the register in the **Reg #** column. You can edit these values.

Hiding and Showing Columns

To hide a column, right-click the column heading and click **Hide Column**. To see it again, right-click any column heading, click **Add Column**, and click the desired column name.

5.15.2. Indirect Data Map Editor

> **To access this editor:**
> Expand the desired controller in the Project pane, then double-click **Indirect Data Map Editor**.
> Or:
> Press Alt+0 to open the Project Pane. Use the arrow keys to highlight **Indirect Data Map**. Press Enter.

Use this editor to view and edit the Indirect Data Map. The Indirect Data Map is intended primarily for certain communication types. See the Indirect Data Map topic for instructions on using the Indirect Data Map.

For general information on using editors, see the Using Editors topic.

Editing the Data Map

Adding registers to the Data Map

- There are two methods of adding a register to the Custom Data Map:
    - Click an unused cell in the Reg # column, click the **...** button and select the desired register from the hierarchy.
    - Or, Click an unused cell in the Reg #, begin typing the file element of the desired register name (ex F12), then select the element from the drop-down list.
- Click the Download button  to apply the changes to the RMC.

Assign a value to a mapped register

To assign a value to a register in the Indirect Data Map:

- Enter the desired value Current column.
- Click the Download button  to apply the changes to the RMC.

Columns

> **Tip:**
> To hide a column, right-click the column heading and click **Hide Column**. To view a hidden column, right-click any column heading, click **Add Column**, and click the desired column name.

Column Name	Description
First Column	The first column is the address of that entry in the Data Map. **Note:** The address in the first column is not the address of the value in the Current column. See the Variable Table topic for details.
Reg #	This is the address of register referenced by the entry in the Indirect Data Map.
Description	The description of the register in the Reg# column.
Units	The units of the register in the Reg# column.
Current	This is the current value of the register in the **Reg #** column. If it is a parameter register, you can edit this value.

5.15.3. Custom Data Map Editor

> **To access this editor:**
> Expand the desired controller in the Project pane, then expand Custom Data Map. Double-click an existing Custom Data Map, or right-click **Custom Data Map** and click **New Custom Data Map** to create a new one.

The Custom Data Map Editor provides an easy method of viewing and editing several registers simultaneously in RMC70Tools. When multiple panes are open in RMC70Tools, it is often difficult to view registers from several editors at the same time. The Custom Data Maps helps solve this problem.

For general information on using editors, see the Using Editors topic.

Editing the Data Map

Adding registers to the Data Map

- There are two methods of adding a register to the Custom Data Map:
    - Click an unused cell in the Reg # column, click the **...** button and select the desired register from the hierarchy.
    - Or, Click an unused cell in the Reg #, begin typing the file element of the desired register name (ex F12), then select the element from the drop-down list.
- Click the Download button  to apply the changes to the RMC.

Edit a mapped register

To assign a value to a register in the Indirect Data Map:

- Enter the desired value Current column.
- Click the Download button  to apply the changes to the RMC.

Columns

> **Tip:**
> To hide a column, right-click the column heading and click **Hide Column**. To view a hidden column, right-click any column heading, click **Add Column**, and click the desired column name.

Column Name	Description
First Column	The first column is the address of that entry in the Data Map. **Note:** The address in the first column is not the address of the value in the Current column. See the Variable Table topic for details.
Reg #	This is the address of register referenced by the entry in the Indirect Data Map.
Description	The description of the register in the Reg# column.
Units	The units of the register in the Reg# column.
Current	This is the current value of the register in the **Reg #** column. If it is a parameter register, you can edit this value.

5.15.4. Using Editors

RMC70Tools has several panes for editing the values in the RMC70. The instructions in this topic apply to the following editors:

- Axes Parameters Pane
- Indirect Data Map Editor
- Variable Table Editor

Tip:
To hide a column in an editor, right-click the column heading and click **Hide Column**. To view a hidden column, right-click any column heading, click **Add Column**, and click the desired column name.

Viewing Project Values and Controller Values

In the editors listed above, there are two types of data: **project data** and **controller data**:

Project data is data in the project file you have created in RMC70Tools. You can save the project data in the RMC70Tool project. Controller data is the data in the RMC70. You can save the controller values in the RMC70 by using the Update Flash (110) command. The project data and controller data may not always be the same. You can edit the project data and download it to the controller. You can also upload data from the controller to the project.

If you are Online and the Auto Refresh checkbox is selected, RMC70Tools always indicates when there is a difference between the controller data and the project data.

Viewing the project values:

To view the project values, click the **Show Project Values** button . If there are no differences between the project values and the controller values, the editor will always be in this mode. You can edit the values in this mode.

Viewing the controller values:

If RMC70Tools has detected a difference between the data in the controller and the data in the project, you can click the **Show Controller Values** button to show the values in the RMC70. If there are no differences, this button is inactive. You cannot edit the controller values in this mode. To change the values in the controller, you must first click the **Show Project Values** button.

Differences between the project values and controller values:

When there is a difference between the project values and the controller values, the symbol will indicate it. The data that is not equal will be highlighted. To make the values equal, you can download or upload. To see what the differing data values are, use the **Show Controller Values** button and the **Show Project Values** button .

Editing Values

Editing the project values:

- Click the **Show Project Values** button to make sure you are in the Show Project Values mode.
- Edit the desired values. See the specific editor topic for details. The links are at the top of this topic.
- To save the project values, on the **File** menu, click **Save**.

Editing the controller values:

- First, edit the project values as described above.
- Click the **Download** button to apply the project values to the RMC70.
- To save the values to Flash, use the Update Flash (110) command.

Uploading the controller values to the project:

- Click the **Upload** button to apply the values from the RMC70 to the project. This is useful when you go online with an RMC70 and discover that the project values are different from the controller values and you want to keep the controller values.

Downloading the project values to the controller:

- Click the **Download** button to apply the values from the project to the RMC70 . This is useful when you go online with an RMC70 and discover that the project values are different from the controller values and you want to apply the project values to the RMC70.

Other Components

Values not Equal

The symbol indicates that the project data and the data in the controller are not equal. The data that is not equal will be highlighted. To apply the controller data to the project, upload the parameters. To apply the project data to the controller, download the parameters.

Auto Refresh Checkbox

The Auto Refresh checkbox specifies whether the data in this window is constantly updated. Selecting this box will turn on the automatic refreshing. Selecting this box may slow down other communications with the RMC70.

Toolbar Buttons

Button	Description
Show Project Values	This buttons shows the values in the *project*. You can edit the values in this mode. When you edit a cell the value in the project changes. To apply the changes to the controller, you must click the download button. If you would like to know what the current values are in the controller, click the **Show Controller Values** button.
Show Controller Values	If RMC70Tools has detected a difference between the data in the controller and the data in the project, you can click this button to show the values in the controller. If there are no differences, this button is inactive. You cannot edit the values in this mode. To edit the values in the controller, you must first click the **Show Project Values** button.
Download	Click this button to download the data from the project to the controller. Use this button when you have edited data and want to apply the changes to the controller. This button is not available in the **Show Controller Values** mode.
Upload	Click this button to upload the data from the controller to the project. Use this button when there are differences between the project and the controller and you want to apply the values in the controller to the project. This button is not available in the **Show Controller Values** mode.

5.15.5. Variable Table Editor

To access this editor:
Expand the desired controller in the Project pane, then double-click **Variable Table**.
Or:
Press Alt+0 to open the Project Pane. Use the arrow keys to highlight **Variable Table**. Press Enter.

Use the Variable Table Editor to create, view, and set the initial value of variables. For instructions on how to use variables in RMC70Tools, see the Variables topic.

For general information on using editors, see the Using Editors topic.

Editing Variables

Note:
To edit the current value of a variable, you must use the Program Monitor.

Create a Variable

- Click an unused **Tag Name** cell. If it contains "0", it is unused.
- Type a name for the variable and press Enter. The variable can now be referenced from anywhere in RMC70Tools using this tag.

Set the Current Value

To edit the current value of a Variable, you must use the Program Monitor.

Set the Initial Value for a Variable

- Click a cell in the **Initial** column.
- Type a value and press Enter.
- Click the Download button to apply the changes to the RMC70.
- The variable will not immediately be set to the default value. Variables are set to the default value when the RMC70 is restarted.

Select a Variable Format

- Click a cell in the Format column.
- Select the desired format and press Enter. The following options exist:
    - Float - can take on any decimal value
    - Integer - can only be whole numbers
    - Bit - can only be 0 or 1

Enter the Units

- Click a cell in the Units column.
- Type the desired units and press Enter. The units are for user reference only and may be anything.

Add a description

- Click a cell in the Description column.
- Type a description and then press Enter.

Columns

Tip:
To hide a column, right-click the column heading and click **Hide Column**. To see it again, right-click any column heading, click **Add Column**, and click the desired column name.

Column Name	Description
First	The first column gives the address of the Current value of the variable

Column in the form F56:*n*, where *n* is the variable number. The address of the Default variable value is F64:*n*, where *n* is the variable number.

Tag Name This is the descriptive name of the variable.

> **Note:**
> When communicating from a PLC, you cannot use Tag Name. You must use the register address.

Units You can enter the units of the variable in this column. The units are only for your reference and have no further importance.

Format You must specify the correct format for successful operations. For example, a variable that is to be compared with an Actual Position must be of the same format as the Actual Position, which is a Float value.

Initial This is the value of the variable before it is modified by a user or a user program. You can edit this value.

Description You can enter a description of the variable in this column.

5.16. Plot Manager

5.16.1. Plot Manager: Viewing Plots

This topic describes the various methods of viewing plots in the Plot Manager. Any of the actions listed below will display a plot. Which plot is plotted is determined in the following manner:

- If the Plot Manager was opened by double-clicking a plot in the Project pane in RMC70Tools, that plot will be plotted.
- Once the Plot Manager has been opened, the highlighted plot in the Plot list will be plotted. Note that it is difficult to determine the plot configuration of each plot in the plot list.

Actions

View the Last RMC70 Plot

To view the latest plot, click the button on the toolbar, or, on the **Online** menu, click **Upload Current Plot**. The latest plot will be uploaded in the Plot pane. The blue Data Captured bar at the top of the Plot pane indicates which portion of the Plot pane is the uploaded plot. Pressing the **Esc** key will stop the upload.

Start and View a Plot

To start a new plot from the Plot Manager, click the button on the toolbar, or, on the **Online** menu, click **Start a New Plot**. A new plot will immediately be started and uploaded to the Plot pane. The blue bar at the top of the Plot pane indicates which portion of the Plot pane is the uploaded plot. Pressing the **Esc** key will stop the upload.

Continuously Capture

- **Start a Continuous Capture**

  To start a continuous capture, click the button on the toolbar, or, on the Online menu, click Re-arm Plot. The Plot Manager will begin continuously capturing plot data. Data will be captured from every loop time of the RMC70, but all the data will not be uploaded to the Plot Manager. As much data as the available communication resources

allow will be uploaded to the Plot Manager. If AutoRefresh is selected in the Status Monitor, Parameter Editor, etc., the number of plot points that can be uploaded will be reduced. The Captured Data bar at the top of the Plot pane indicates the captured samples with a blue section.

- **Stop a Continuous Capture**

  To stop a continuous capture, click the Stop Continuous Capture button on the toolbar, or, on the Online menu, click Stop Capture. The continuous capture will stop, and the plot data will be uploaded and will fill in the empty samples in the plot. The Plot Length determines the length of data that is uploaded and filled in.

  > **Note:**
  > The filling in of data may begin to the left of the visible area of the Plot pane. To view the data filling in, scroll to the left.

Manually Trigger and View a Plot

Before triggering a plot, it must first be re-armed. To re-arm a plot, click the button on the toolbar, or, on the **Online** menu, click **Re-arm Plot**.

To trigger a plot, click the button on the toolbar, or, on the **Online** menu, click **Trigger Plot**. The plot data will be uploaded starting with the sample determined by the trigger settings in the Plot Properties. The length of uploaded data is the plot length.

> **Note:**
> The data may begin uploading to the left of the visible area of the Plot pane. To view the data filling in, scroll to the left.

5.16.2. Plot Manager Overview

> **To access the Plot Manager:**
> On the **Tools** menu, click **Plot Manager** and it will open.
> Or, if you are online, in the Project Pane, expand the **Plots** folder and double-click a plot. The plot you clicked on will upload from the RMC70. If there was no last plot, the plot will continuously capture data from the RMC70.

Use the Plot Manager in RMC70Tools to view, start, trigger, save and print plots. For general details on using plots in the RMC70, see the Plots Overview topic. For advanced plotting capabilities of the RMC70, see the Advanced Plot Capabilities topic.

For help on the various parts of the Plot Manager, see the Plot Manager Elements topic.

Viewing Plots

How to...	
Upload a Plot	To view the last plot, click the Upload Plot button, or on the **Online** menu, click **Upload Current Plot**. The last plot will upload into the Plot pane. If there is no last plot, the plot will continuously capture.
Start a Plot	To start a plot, click the Start Plot button. A plot will start immediately on the RMC70 and will upload into Plot pane.
Trigger a Plot	To trigger a plot, click the Trigger Plot button.

Continuously Capture Data	To continuously capture a plot, click the Re-arm Plot button.
Stop Continuous Capture	To stop continuously capturing a plot, click the Stop Continuous Capture button.

Managing the Plot List

The Plot List pane lists all the plots you have uploaded from the RMC. In this pane, you can save plots, delete plots, and print plots.

Toolbar Buttons

Button	Command	Equivalent Menu Action
	New	On the **File** menu, click **New**.
	Open	On the **File** menu, click **Open**.
	Save Plot	On the **File** menu, click **Save**.
	Delete Plot	On the **Edit** menu, click **Delete Plot**.
	Switch to Graph View	On the **View** menu, click **Graph View**.
	Upload Current Plot from Controller	On the **Online** menu, click **Upload Current Plot**.
	Start Plot	On the **Online** menu, click **Start a New Plot**.
	Stop Capturing Plot	On the **Online** menu, click **Stop Capture**.
	Trigger Plot	On the **Online** menu, click **Trigger Plot**.
	Re-arm Plot	On the **Online** menu, click **Re-arm Plot**.

5.16.3. Plot Manager Elements

The Plot Manager is divided into the following elements:

Plot List Pane

The Plot List pane displays a list of all plots in the project. Every plot that is uploaded from the module is automatically saved under the Recently Uploaded Plots folder in this list. The following actions can be done in the Plot List:

- **Delete a Plot**
  The currently highlighted plot in the Plot List will be deleted.
- **Save Plots**
  The currently highlighted folder in the Plot List will be saved to a file with an .rmcplots extension. The folder will also remain in the Plot List but will be renamed.
- **View a Plot**
  Double-click a plot in the Plot List to view it. The plot will appear in the Plot pane.

Detail Pane

The Detail pane has an associated hairline cursor in the Plot pane. The Detail pane displays the plot data at the sample at which the hairline cursor is located. The color of

each data item in the Detail pane corresponds to the color of each line in the plot. To change the data to be plotted, see the Selecting Data to Plot topic.

Plot Pane

The Plot pane displays the data in the plot. The data can be displayed in Graph View or Spreadsheet View:

- **Graph View**
  The Graph view displays the plot data in line graph form, as a function of time. Each register is plotted as different color. To select the Graph view, click the button in the toolbar.
    - **Hairline Cursor**
      Move the hairline cursor to view the data at any sample. To move the hairline cursor, use the left and right arrow keys, Page Up, Page Down, or click and drag with the mouse.
    - **Captured Data Bar**
      The Captured Data bar is located at the top of the Plot pane. This bar is blue at every sample that in the Plot pane that contains data captured from the RMC70.
- **Spreadsheet View**
  The Spreadsheet view displays the plot data in numerical form.

Toolbar

The toolbar holds buttons for commonly used commands. To help identify these buttons, hover the pointer over a button and a ToolTip will pop up and a description will be displayed on the status bar.

For details on each toolbar button, see this list:

Button	Command	Equivalent Menu Action
	New	On the **File** menu, click **New**.
	Open	On the **File** menu, click **Open**.
	Save Plot	On the **File** menu, click **Save**.
	Delete Plot	On the **Edit** menu, click **Delete Plot**.
	Switch to Graph View	On the **View** menu, click **Graph View**.
	Upload Current Plot from Controller	On the **Online** menu, click **Upload Current Plot**.
	Start Plot	On the **Online** menu, click **Start a New Plot**.
	Stop Capturing Plot	On the **Online** menu, click **Stop Capture**.
	Trigger Plot	On the **Online** menu, click **Trigger Plot**.
	Re-arm Plot	On the **Online** menu, click **Re-arm Plot**.

6. Modules

6.1. CPU Modules

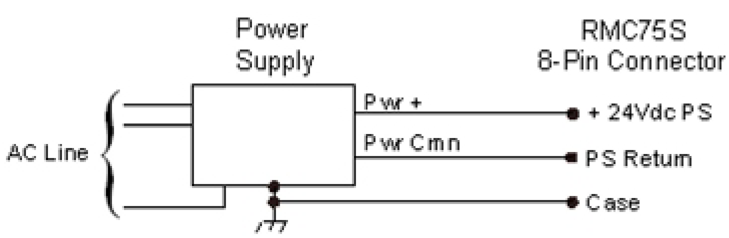
6.1.1. RMC75S

6.1.1.1. RMC75S CPU Module

6.1.1.2. RMC75S Wiring

The RMC75S CPU module contains a connector for power and RS-485 communications, and two connectors for RS-232 communications.

Wiring Power



Wiring the Communications

For wiring RS-232 communications and the RS-232 Monitor Port, see the RS-232 Wiring for the RMC70 topic. For wiring RS-485, see the RS-485 Wiring topic.

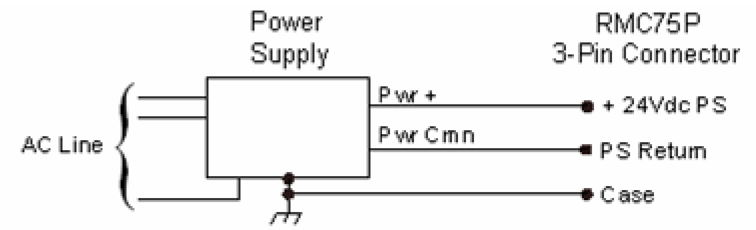
6.1.2. RMC75P

6.1.2.1. RMC75P CPU Module

6.1.2.2. RMC75P Wiring

The RMC75P CPU module contains a connector for power, and two connectors for communications.

Wiring Power



Wiring the Communications

For wiring the RS-232 Monitor Port, see the RS-232 Wiring for the RMC70 topic.

6.1.3. CPU Modules Overview

A CPU Module contains the processing unit and communications of the RMC70 motion controller. The CPU module, together with an Axis module attached to the front of the CPU Module, make up a complete motion controller, called a Base module. Optional Expansion modules can be added to the right of the motion controller.

The front panel of the CPU contains status LEDs and connectors for power and communications.

The following RMC70 CPU Modules are available:

Module	Communications Type
RMC75S	Serial
RMC75P	PROFIBUS

See also:

Physical Components Overview

RMC70 Part Numbering

6.1.4. CPU Module LEDs

Every RMC70 CPU module has a **Controller** LED and one or more communication LEDs. These LEDs reflect the operational status of the CPU and communications as described in this topic.

LEDs

Controller LED

This bi-color (red/green) LED has the following states:

State	Description
Steady Off	No power.
Steady Green	RUN Mode
Flashing Green (Slow)	PROGRAM Mode
Flashing Green (Fast)	Updating Flash
Flashing Red	The device is in the loader. This should occur briefly when the controller is powered up. If this occurs during normal operation, an error has occurred in the controller and it must be reset. Cycle power to the RMC70 to reset it.
Steady Red	Non-Recoverable major fault. That is, the module cannot boot at all, not even into the loader. The module will likely need to be shipped back to Delta for repair. Some things to try to get it into the Loader (to be done at the direction of technical support) include removing all expansion and axis modules, and setting the Force to Loader jumper under the axis module.

Communication LEDs

These LEDS are located below the Controller LED on a gray background. Click the communication type below for details on the LEDs:

Serial LEDs on the RMC70S

Transmit LED

This LED reflects when data is being transmitted on the second serial port (RS-232/485).

State	Description
Steady Off	No power or no data being transmitted.
Steady or Flickering Green	Data is being transmitted.

Receive LED

This LED reflects when data is being received on the second serial port (RS-232/485).

State	Description
Steady Off	No power or no data being received.
Steady or Flickering Green	Data is being received.

PROFIBUS LEDs on the RMC70P

NET LED

State	Description
Steady Off	The RMC's PROFIBUS channel is not communicating control data with the PROFIBUS master. There are a number of reasons why this might be the case including: no power, no PROFIBUS cable, invalid PROFIBUS cable, improper PROFIBUS cable termination, no PROFIBUS master connected, improper configuration in the PROFIBUS master including wrong address, parameters, or configuration, and the PROFIBUS master has not been told to go on-line. The RMC70 Event Log can help narrow down the problem.

Steady Green	The RMC's PROFIBUS channel is properly communicating control data with the PROFIBUS master.

NOTE:
These are the only two LED states, but it is possible to have the Net LED flashing or flickering green, but this indicates that the RMC is going on- and off-line on the PROFIBUS channel and generally indicates a network or configuration problem.

6.2. Axis Modules

6.2.1. Axis Modules Overview

An Axis Module is the part of the RMC70 motion controller that interfaces to transducers and axis drive. The axis module is mounted on the front of the RCM70 Control Module (see the Physical Components Overview topic for an image). It has status LEDs and connectors for the axis wiring. Each axis module interfaces to certain types of transducers and certain types of Control Outputs.

Axis Modules all have a Control Output. For other module types without a Control Output, such as analog input, digital I/O, and quadrature input, see the Expansion Modules. For a description of the various module types that make up the RMC70, see the Physical Components Overview topic.

Available Modules

The following Axis Modules are available for the RMC70:

(the number following MA indicates the number of axes)

Module	Transducer Input Type	Output
MA1	MDT or SSI	Analog
MA2	MDT or SSI	Analog

For details on part numbers, see the RMC70 Part Numbering topic.

6.2.2. Control Output

Each control axis has one Control Output. The Control Output is the voltage supplied to drive the amplifier, valve, or drive for the system to be controlled. It is an analog output with a voltage range of -10 V to + 10 V.

Configuring the Control Output

The user can configure the polarity of this output with the Invert Output Polarity parameter. To invert the polarity of the Control Output, set this bit.

Modules with a Control Output

The following Axis modules have a Control Output:

- MA
- AA

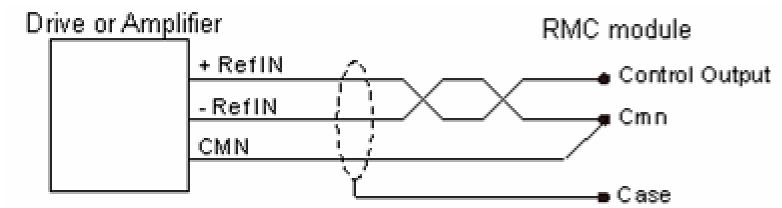
Wiring

The valve or motor drive connects to the following pins:

Pin	Function
Control Output	Control Output
Cmn	Drive Common. Each axis has two Cmn pins. Either of the 2 pins may be used.

Note:
The Drive polarity can be set with the Invert Output Polarity parameter.

Drive Wiring Diagram



6.2.3. Enable Output

Each control axis has one Enable Output. It is intended to be connected to a drive amplifier, allowing the RMC to turn it on or off with the Set Enable Output (67) command.

Configuring the Enable Output

The user can configure the behavior of this output with the Enable Output Behavior parameter. The parameter has the following two choices:

- **Active Closed** - When the Enable Output is set, the Enable output switch is closed.
- **Active Open** - When the Enable Output is set, the Enable output switch is opened.

Modules with a Enable Output

The following Axis modules have a Enable Output:

- MA
- AA

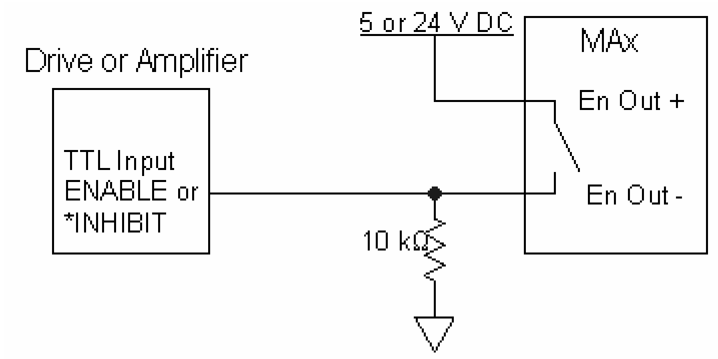
Wiring

The Enable output is a solid state relay (SSR). When it is "off", it has high impedance, and when on it has low impedance (10 Ω maximum). Because the Enable output is isolated, the user must power it externally. The maximum current and voltage for the Enable output is 100 mA and 30 V.

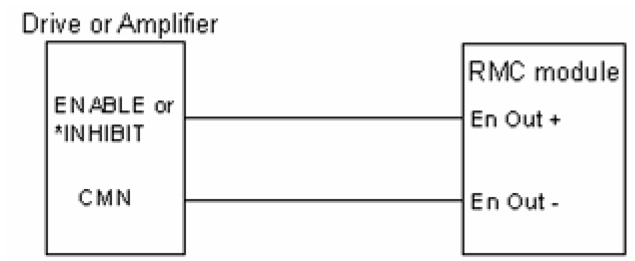
The Enable output has a + and - connection. Both lines must be connected for the output to function. Because both sides of the output are provided, the switches may be independently connected in a high side or low side configuration (that is, with the load (input) on the source or sinking side of the output). See the wiring diagrams below.

Enable Output Wiring Diagrams

To TTL input (high = enable):



To active low Enable input:



6.2.4. Fault Input

Each control axis has one Fault input. It is intended to be connected to the drive amplifier or other source for a safety interlock. When the Fault input is active, the Fault Error bit will be set. This will cause a halt if the Auto Stops are configured to do so, and the axis is not in the Direct Drive state.

Configuring the Fault Input

The user can select the active state of this input with the Fault Input Polarity parameter. The parameter has the following two choices:

- **Active High** - When the voltage applied to the Fault Input is greater than 6 volts, the Fault input is active.
- **Active Low** - When the voltage applied to the Fault Input is less than 6 volts, the Fault input is active.

Modules with a Fault Input

The following Axis modules have a Fault Input:

- MA
- AA

Wiring

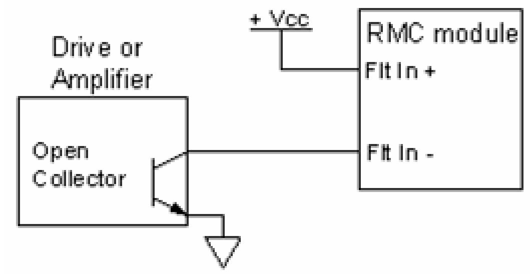
The Fault input is compatible with signal levels ranging from 12V to 24V. The Fault Input draws 2.7mA maximum and becomes active at 6V.

> **NOTE:**
> The Fault input turns on when a current flows. The polarity of the voltage is unimportant.

Fault Input Wiring Diagrams

The following are some Fault input wiring diagrams:

From Open Collector Output:



6.2.5. AA

6.2.5.1. AA Module

The AA module is one of the axis modules available for the RMC70. It interfaces to analog voltage or current transducers and has 1 analog servo output per axis. The AA module is available with either 1 or 2 axes, called the AA1 and the AA2.

Transducer Types

Each axis of the AAx axis module interfaces to any of the following transducers:

- **Voltage**
  -10V to +10V
- **Current**
  4-20mA

Control Types

The following types of control are possible with the MA module.

- **Position**
- **Velocity** - currently not available
- **Position/Pressure** - this requires an AP2 module

Specifications

See the AA Specifications topic.

Setup

To set up the AA module, read the following topics:

AA Wiring

AA Scaling

Other

AA LEDS

6.2.5.2. AA Axis Module Specifications

For general specifications on the RMC70, see RMC70 Specifications.

General

Axes	One or Two per module

Transducer Interface

Note:
Each axis is individually selectable for Voltage or Current.

Analog Interface

Inputs	Two differential per axis
Input Ranges	-10V to +10V

Control Outputs

Range	±10 V @ 5 mA (2 kW or greater load)(For current drive, use the VC2100 accessory:±10 mA to ±200 mA in 10 mA steps)
Tolerance At 10 V:	Currently unavailable
Resolution	16 bits
Output Isolation	Not isolated
Overload protection	One-second short-circuit duration
Overvoltage protection	Outputs are protected by clamp diodes

6.2.5.3. AA Wiring

The AA Axis module can be wired to voltage or current feedback transducers. The AA module also a Fault input, Enable output, and a Control Output. If the AA module has two axes, each axis' pin-out is identical.

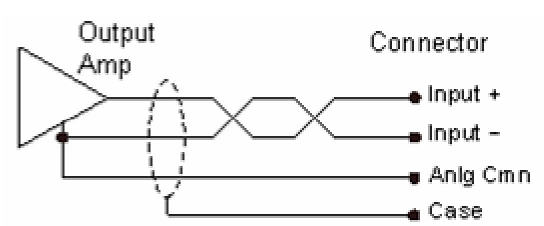
Use shielded twisted pairs for all connections to inputs and outputs. Route the transducer wiring separate from other wiring. You must provide the power supplies needed by your transducers. See Wiring Guidelines for general wiring information.

NOTE:
The example schematics do not include transducer pin numbers, color codes, or power supply requirements, since these vary between different transducers. To determine your power supply needs and connector pin-outs or cable color codes, consult your transducer manufacturers documentation.

Wiring Diagrams and Instructions

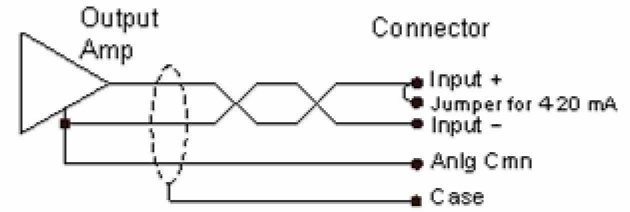
Voltage Feedback Transducers

Voltage feedback transducers can be connected directly to the **Input +** and **Input -** connections of the desired axis. The following configuration is recommended:



Current Feedback Transducers

Current feedback transducers are connected in the same way as voltage transducers except that a jumper must be inserted between the **Input+** and **Jumper for 4-20 mA** pins. The label indicates where this jumper should be connected. This places a resistor internal to the RMC across the two inputs, thus converting the current to a voltage input. The following wiring diagram shows a suggested configuration:



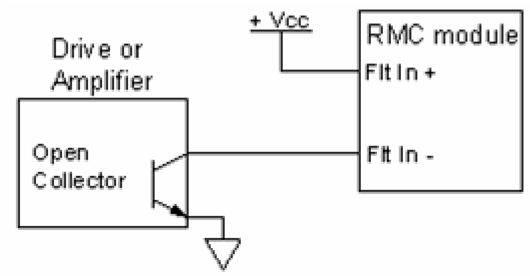
Fault Input

The Fault input is compatible with signal levels ranging from 12V to 24V. The Fault Input draws 2.7mA maximum and turns on at 6V. The Fault input turns on when a current flows. The polarity of the voltage is unimportant. See the Fault Input topic for more details.

Fault Input Wiring Diagrams

The following are some Fault input wiring diagrams:

From Open Collector Output:



Enable Output

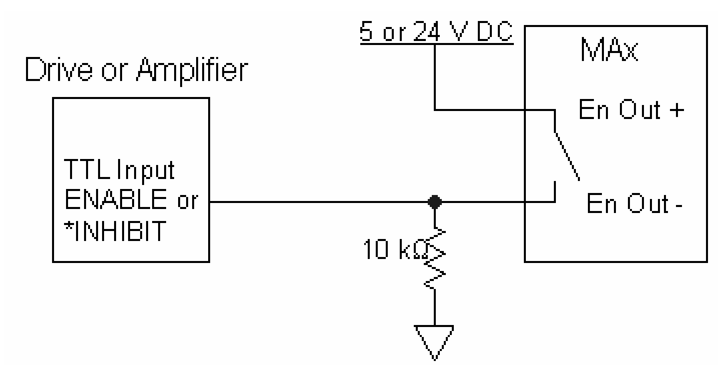
The Enable output is a solid state relay (SSR). When it is "off", it has high impedance, and when on it has low impedance (10 Ω maximum). Because the Enable output is isolated, the user must power it externally. The maximum current and voltage for the Enable output is 100 mA and 30 V.

The Enable output has a + and - connection. Both lines must be connected for the output to function. Because both sides of the output are provided, the switches may be independently connected in a high side or low side configuration (that is, with the load (input) on the source or sinking side of the output). See the wiring diagrams below.

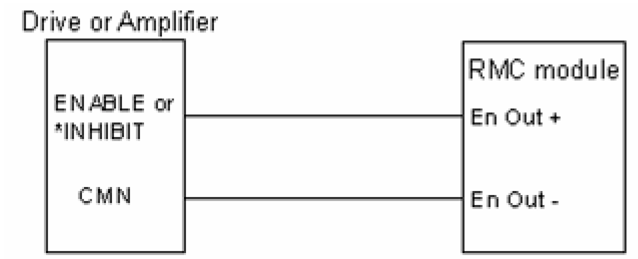
See the Enable Output topic for more details on the Enable Output.

Enable Output Wiring Diagrams

To TTL input (high = enable):



To active low Enable input:



Control Output

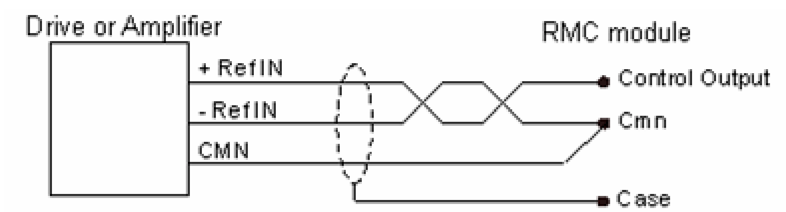
The valve or motor drive connects to the following AA*x* pins:

Pin	Function
Control Output	Control Output
Cmn	Control Output Common. Each axis has two Cmn pins. Either of the 2 pins may be used.

See the Control Output topic for more details on the Control Output.

Note:
The Control Output polarity can be set with the Invert Output Polarity parameter.

Control Output Wiring Diagram



6.2.5.4. AA LEDs

The AA modules has 2 LEDs per axis:

Axis LED
This LED represents the input and control status for the axis. This bi-color (red/green) LED has the following states:

State	Description
Steady Off	No power or not configured for use.
Steady Green	The axis is either in closed loop control or is a reference axis.
Flashing Green	This input in open loop (not possible for reference axes).
Flashing Red	A Closed Loop Halt or External Halt has occurred because an error bit is set.
Steady Red	An Open Loop Halt or Disable Output has occurred because an error bit is set.

En/F LED

This LED represents the status of the Enable output and Fault input for the axis. This bi-color (red/green) LED will have the following states:

State	Description
Steady Off	No power or not configured for use.
Steady Green	The Fault input is inactive, and the Enable output is active.
Flashing Green	The Fault input is inactive, and the Enable output is inactive.
Steady Red	The Fault input is active. Notice that it is not possible to tell if the Enable output is on or off.

6.2.5.5. Analog Scaling for the AA Module

To have any useful meaning, the Voltage from the analog transducer must be scaled to position units. This topic describes how to correctly calculate the Scaling, Offset and Invert Feedback Polarity parameters for an analog position or velocity transducer.

Scaling Voltage to Position Units

The RMC70 calculates the Actual Position every control-loop time using the following formula:

Actual Position = (Voltage x Position Scale) + Position Offset

To correctly scale the axis for your particular application, use the method below to find the Position Scale and Position Offset parameters.

Method: P_0 / P_1 Calculation

The accuracy of this method depends on how accurately you can measure two positions of the axis.

1. Physically measure the axiss position at two points and record the value of the counts register at each point. Call the *smaller* measured position P_0, and its corresponding counts C_0. Call the *greater* measured position P_1, and its corresponding counts C_1.
2. Calculate the Position Scale with the following equation:

   Position Scale = $(P_0-P_1)/(C_0-C_1)$
3. Calculate the Position Offset with the following equation:

   Position Offset = P_0 - Position Scale x C_0

Scaling Voltage to Velocity Units for Tachometers

The RMC70 calculates the Actual Velocity from the tachometer voltage every control-loop time using the following formula:

Actual Velocity = (Voltage x Velocity Scale) + Velocity Offset

To correctly scale the axis for your particular application, use the method below to find the Velocity Scale and Velocity Offset parameters.

Method: S_0 / S_1 Calculation

1. Measure the axiss velocity at two different velocities and record the value of the Voltage register at each point. Call the *smaller* measured velocity S_0, and its corresponding Voltage V_0. Call the *greater* measured velocity S_1, and its corresponding Voltage V_1.

2. Calculate the Velocity Scale with the following equation:

$$\text{Velocity Scale} = (S_0\text{-}S_1)/(V_0\text{-}V_1)$$

3. Calculate the Velocity Offset with the following equation:

$$\text{Velocity Offset} = S_0 - \text{Velocity Scale} \times V_0$$

6.2.6. MA

6.2.6.1. MA Axis Module

The MA module is one of the axis modules available for the RMC70. It interfaces to MDT and SSI transducers and has 1 analog servo output per axis. The MA module is available with either 1 or 2 axes, called the MA1 and the MA2.

Transducer Types

Each axis of the MAx axis module interfaces to any of the following transducers:

- **MDT** - Magnetostrictive Displacement Transducers
- **SSI** - Synchronous Serial Interface

These transducers have many different options. For a complete list of the transducer options supported by the MA module, see the Supported Transducer Options topic.

Control Types

The following types of control are possible with the MA module.

- **Position**
- **Velocity** - currently not available
- **Position/Pressure** - this requires an AP2 module

Specifications

See the MA Specifications topic.

Setup

To set up the MA module, read the following topics:

MA Wiring

MA MDT Scaling

Other

MA LEDs

6.2.6.2. MAx Supported Transducer Options

Transducer Options

Many different versions of MDT and SSI transducers exist, and the RMC70 MA axis module supports a very wide variety of them. The following table summarizes the major transducer options supported by the MA module:

MDT Options	Value

Types	Start/Stop (S/S) Pulse-Width-Modulated (PWM)
Count Range	32 bits
Wiring	Differential, 5V, RS-422
Interrogation Modes	Local, Remote
MDT Count Rate	240 MHz
Output Resolution	16 bits

SSI Options	Value
SSI Data Bits	8 to 32
SSI Format	Binary or Gray Code
SSI Errors	None, zero, or bit20
Clock Rates	150, 250, and 375 kHz

> **Note:**
> If you are using SSI, it should be the synchronized type. This ensures that the time between position samples matches the control loop time of the RMC. If the transducer is not synchronized, the sample time may not match and make precise speed control difficult.

Configurations

The inputs to the MA module can be configured as summarized in the following table:

Transducer Interface	Input Mode	Absolute/ Incremental	Orientation
MDT	Position	Absolute	Linear
SSI	Position	Absolute	Linear
	Position	Absolute	Rotary
	Position	Incremental	Linear
	Position	Incremental	Rotary

6.2.6.3. MA Axis Module Specifications

For general specifications on the RMC70, see RMC70 Specifications.

General

Axes	One or Two per module

Transducer Interface

> **Note:**
> Each axis is individually selectable for MDT or SSI.

MDT Interface

Inputs	Two RS422 differential supported	Note: single-ended is not
Outputs	Two RS422 differential supported	Note: single-ended is not
ESD protection	15kV	
Circulations	None	
Resolution	0.0005" Start/Stop; 0.00005" PWM	
Count Rate	240MHz	

SSI Interface

Inputs	Two RS422 differential
Clock Outputs	Two RS422 differential
Clock Frequency	Software selectable 150, 250, or 375 kHz.
Cable Type	Twisted pair, shielded
Cable Length Maximum	Currently unavailable
ESD protection	15kV
Resolution	Transducer dependent. Down to 1 µm (approx. 0.00004")
Count Encoding	Binary or Gray Code
Count Data Length	4 to 32 bits

Control Outputs

Range	±10 V @ 5 mA (2 kW or greater load)(For current drive, use the VC2100 accessory:±10 mA to ±200 mA in 10 mA steps)
Tolerance At 10 V:	Currently unavailable
Resolution	16 bits
Output Isolation	Not isolated
Overload protection	One-second short-circuit duration
Overvoltage protection	Outputs are protected by clamp diodes

6.2.6.4. MAx Wiring

The MA*x* module can be wired to MDT and SSI transducers. Each axis on the MAx also has a Fault input, Enable output, and a Drive output. If the MA module has two axes, each axis' pinout is identical.

Use shielded twisted pairs for all connections to inputs and outputs. Route the transducer wiring separate from other wiring. You must provide the power supplies needed by your transducers. See Wiring Guidelines for general wiring information.

NOTE:
The following example schematics do not include transducer pin numbers, color codes, or power supply requirements, since these vary between different transducers. To determine your power supply needs and connector pin-outs or cable color codes, consult your transducer manufacturers documentation.

Wiring Diagrams and Instructions

MDT Transducers

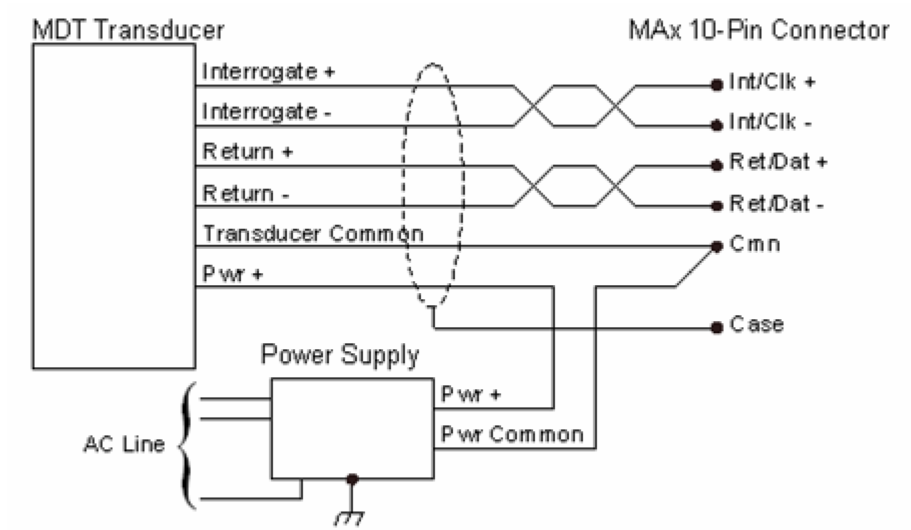
The RMC70 only interfaces to transducers with Differential Line Driver (RS422) signals. Single-ended (TTL) transducers are *not* supported due to low noise immunity. MDT transducers connect to the following MA*x* pins:

Pin	Function	Additional Possible Manufacturer Designation
Int/Clk +	MDT + Interrogation	Interrogate + Input
Int/Clk -	MDT - Interrogation	Interrogate - Input
Cmn	MDT Common	
Ret/Dat +	MDT + Return	Pulse + Output
Ret/Dat -	MDT - Return	Pulse - Output
Case	RMC70 Chassis	

Wiring instructions:

- Connect the '+Int' and '-Int' between the transducer and the MA module for the interrogation signal.
- Connect the '+Ret' and '-Ret' between the transducer and the MA module for the return signal.
- Connect the transducer DC ground to Cmn. Each axis has two Cmn pins. Either of the 2 pins may be used.
- The transducer power supply is not provided by the RMC. The user must supply a power supply. The power supply common may be connected to the same common that is connected to the RMC70 CPU Cmn.

Diagram



SSI Transducers

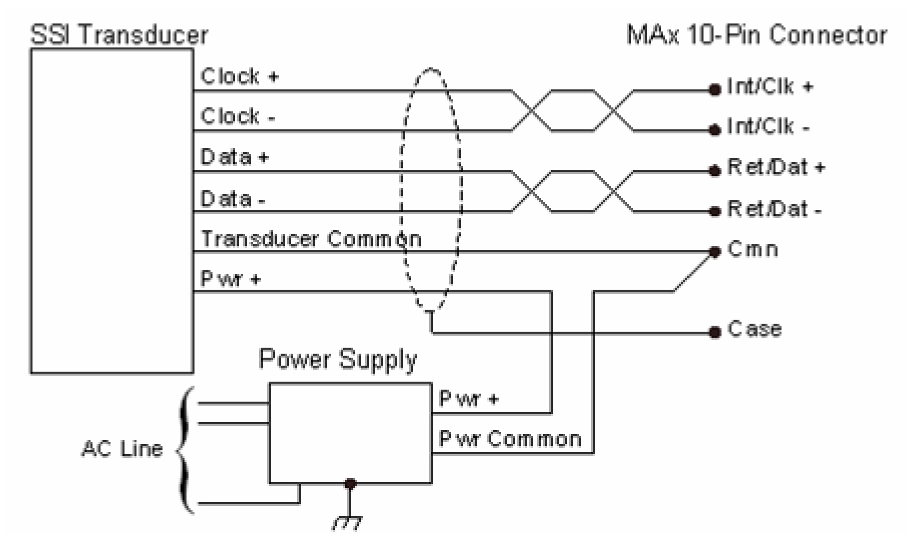
SSI uses differential line driver (RS422) clock and data signals. Connect the transducer DC ground to SSI Cmn. SSI transducers connect to the following MA*x* pins:

Pin	Function
Int/Clk +	SSI + Clock
Int/Clk -	SSI - Clock
Cmn	Common
Ret/Dat +	SSI + Data
Ret/Dat -	SSI - Data
Case	RMC70 Chassis

Wiring Instructions

- Connect both the +Clock and -Clock between the MA module and transducer for the clock signal.
- Connect both the +Data and -Data between the MA module and transducer for the data signal.
- Connect the transducer DC ground to Cmn. Each axis has two Cmn pins. Either of the 2 pins may be used.
- The transducer power supply is not provided by the RMC. The user must supply a power supply. The power supply common may be connected to the same common that is connected to the RMC70 CPU Cmn.

Diagram



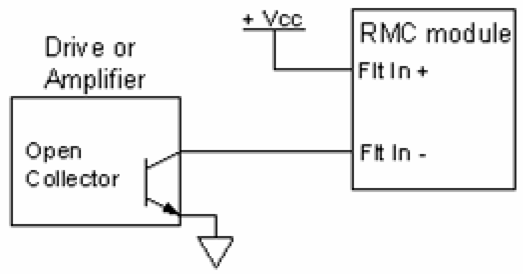
Fault Input

The Fault input is compatible with signal levels ranging from 12V to 24V. The Fault Input draws 2.7mA maximum and turns on at 6V. The Fault input turns on when a current flows. The polarity of the voltage is unimportant. See the Fault Input topic for more details.

Fault Input Wiring Diagrams

The following are some Fault input wiring diagrams:

From Open Collector Output:



Enable Output

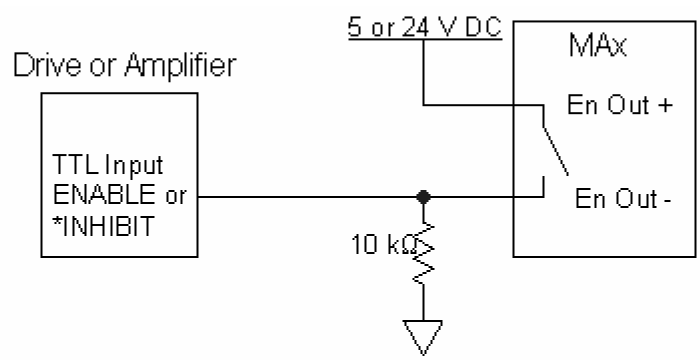
The Enable output is a solid state relay (SSR). When it is "off", it has high impedance, and when on it has low impedance (10 Ω maximum). Because the Enable output is isolated, the user must power it externally. The maximum current and voltage for the Enable output is 100 mA and 30 V.

The Enable output has a + and - connection. Both lines must be connected for the output to function. Because both sides of the output are provided, the switches may be independently connected in a high side or low side configuration (that is, with the load (input) on the source or sinking side of the output). See the wiring diagrams below.

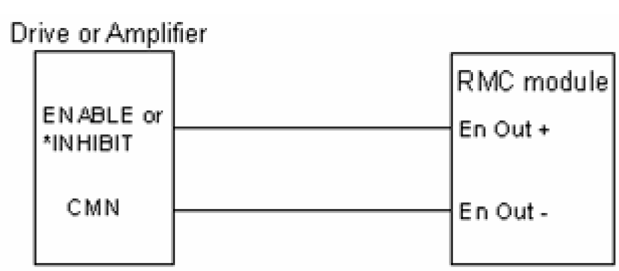
See the Enable Output topic for more details on the Enable Output.

Enable Output Wiring Diagrams

To TTL input (high = enable):



To active low Enable input:



Control Output

The valve or motor drive connects to the following MA*x* pins:

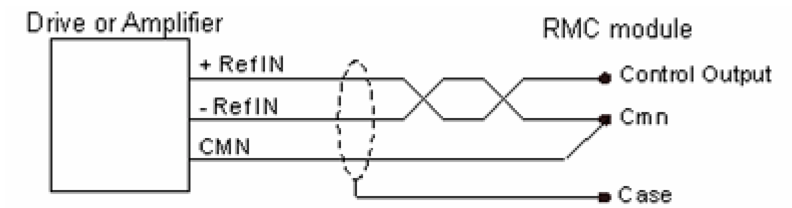
Pin	Function
Control Output	Control Output
Cmn	Control Output Common. Each axis has two Cmn pins. Either of the

2 pins may be used.

See the Control Output topic for more details on the Control Output.

Note:
The Control Output polarity can be set with the Invert Output Polarity parameter.

Control Output Wiring Diagram



6.2.6.5. MA LEDs

The MA modules has 2 LEDs per axis:

Axis LED
This LED represents the input and control status for the axis. This bi-color (red/green) LED has the following states:

State	Description
Steady Off	No power or not configured for use.
Steady Green	The axis is either in closed loop control or is a reference axis.
Flashing Green	This input in open loop (not possible for reference axes).
Flashing Red	A Closed Loop Halt or External Halt has occurred because an error bit is set.
Steady Red	An Open Loop Halt or Disable Output has occurred because an error bit is set.

En/F LED
This LED represents the status of the Enable output and Fault input for the axis. This bi-color (red/green) LED will have the following states:

State	Description
Steady Off	No power or not configured for use.
Steady Green	The Fault input is inactive, and the Enable output is active.
Flashing Green	The Fault input is inactive, and the Enable output is inactive.
Steady Red	The Fault input is active. Notice that it is not possible to tell if the Enable output is on or off.

6.2.6.6. MDT Scaling

To have any useful meaning, the counts from the transducer must be scaled to position units. The Position Scale, Position Offset and Invert Feedback Polarity parameters define the position units as a function of transducer counts. This topic describes how to correctly calculate these parameters for an MDT transducer.

Scaling Counts to Position Units

The RMC70 calculates the Actual Position every control-loop time using the following formula:

Actual Position = (Counts x Position Scale) + Position Offset

> **Note:**
> If the Actual Position filter is applied, the RMC filters the Actual Position after calculating it with the above formula.

To correctly scale the axis for your particular applications, use the methods below to find the Position Scale and Position Offset parameters.

Determining the Scale and Offset

There are two methods of finding the correct Position Scale and Position Offset.

Method 1: P_0 / P_1 Calculation

The accuracy of this method depends on how accurately you can measure two positions of the axis.

1. Physically measure the axis position at two points and record the value of the counts register at each point. Call the *smaller* measured position P_0, and its corresponding Counts C_0. Call the *greater* measured position P_1, and its corresponding Counts C_1.
2. Calculate the Position Scale with the following equation:

   Position Scale = $(P_0-P_1)/(C_0-C_1)$
3. Calculate the Position Offset with the following equation:

   Position Offset = P_0 - Position Scale x C_0

Method 2: Using the MDT Calibration Number

1. Obtain the following information (usually from the transducer label or data sheet):
   - Calibration Number (calibration constant) in µs/in or µs/µm (usually about 9.012 µs/in)
   - Number of Recirculations (if it has no recircs, use a value of 1)
2. Calculate the Position Scale with the following formula:

$$\text{Position Scale (pu/counts)} = \frac{1}{\text{Calibration Number }(\mu s/pu) \times 240\text{Mhz} \times \#\text{ of Recircs}}$$

   where
   - "pu" is the position units, e.g. in, mm, m, etc. If you want position units other than what is given by the Calibration Number (e.g. in,mm, etc.), you must convert the Calibration Number to the units you want.
   - The 240MHz value comes from the RMCs internal counter. If you change the units from the above equation, you make sure the untis for the equation work out properly. The units for MHz is [10^6/sec].
3. If you wish to reverse the direction of the feedback, make the Scale negative.
4. Calculate the Position Offset:
   - On the RMC70Tools toolbar, click the **Go Online** button ().
   - On the RMC70Tools toolbar, click the **Axis Tools** button ().
   - In the **Axis Tools**, in the **Axis Status Registers** pane, click the **All** tab, and expand the **Feedback** section.
   - Look at the **Actual Position** register and the **Counts** register.

- Move the axis to a point where you know what the **Actual Position** register should be (such as 0). Call this position P_0. Record the **Counts** at that position and call that value C_0.

- Calculate the Position Offset using the following equation:

Position Offset = P_0 - Position Scale x C_0

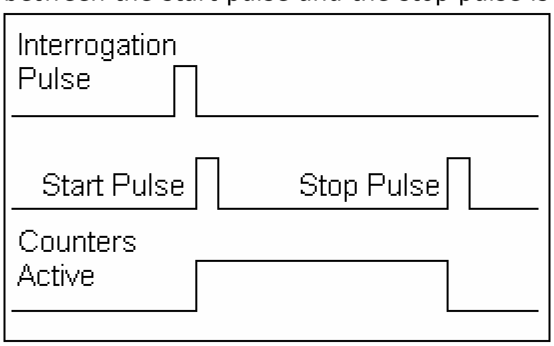
6.2.6.7. MDT Transducer Fundamentals

Magnetostrictive Displacement Transducers (MDT) are designed for use in rugged industrial environments. They are non-contact, wear-free, highly reliable, and offer accurate and repeatable linear position measurement. In the motion control industry, magnetostrictive displacement transducers are typically inserted into hydraulic cylinders for measurement of the cylinders position.

Each axis on the RMC70 MA axis modules can be individually configured for MDT or SSI inputs. Each axis configured for magnetostrictive transducers can be configured for a Start/Stop transducer or a Pulse Width Modulated transducer.

Start / Stop

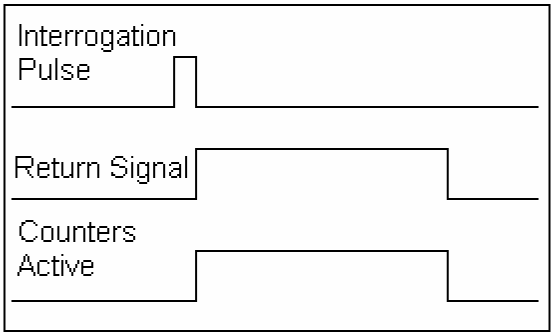
To make a measurement with a Start/Stop transducer, the RMC sends an interrogation pulse to the transducer. The transducer responds by returning 2 pulses—a Start pulse and a Stop pulse. The RMCs internal counters begin to count when the first pulse, Start, is received and stop counting when the second pulse, Stop, is received. The time between the start pulse and the stop pulse is proportional to the transducer position.



Start/Stop Pulse Transducer

PWM

To make a measurement with a Pulse Width Modulated transducer, the RMC sends an interrogation pulse to the transducer. The transducer responds with a return signal. The return signal is high while the transducer is determining its position. The RMC counts during the time that the return signal is high. That time that the return signal is high is proportional to the transducer position.



Pulse Width Modulated Transducer

The value obtained from the PWM or Start/Stop counter is put in the Raw Counts register for that axis. The Raw Counts are converted to Counts and then into an Actual Position in user-defined units.

6.2.6.8. SSI Scaling

To have any useful meaning, the counts from the transducer must be scaled to position units. The Position Scale, Position Offset and Invert Feedback Polarity parameters define the position units as a function of transducer counts. This topic describes how to correctly calculate these parameters for an SSI transducer.

Scaling Counts to Position Units

The RMC70 calculates the Actual Position every control-loop time using the following formula:

Actual Position = (Counts x Position Scale x Feedback Polarity) + Position Offset

where

Feedback Polarity = 1 if the Invert Feedback Polarity parameter is not selected, -1 if it is selected.

Note:
If the Actual Position filter is applied, the RMC filters the Actual Position after calculating it with the above formula.

To correctly scale the axis for your particular applications, use the methods below to find the Position Scale and Position Offset parameters.

Determining the Scale and Offset

There are two methods of finding the correct Position Scale and Position Offset.

Method 1: P_0 / P_1 Calculation

The accuracy of this method depends on how accurately you can measure two positions of the axis.

1. Physically measure the axis position at two points and record the value of the counts register at each point. Call the *smaller* measured position P_0, and its corresponding Counts C_0. Call the *greater* measured position P_1, and its corresponding Counts C_1.
2. Calculate the Position Scale with the following equation:

   Position Scale = $(P_0-P_1)/(C_0-C_1)$
3. Calculate the Position Offset with the following equation:

   Position Offset = P_0 - Position Scale x C_0

Method 2: Using the Transducer Resolution

Linear Transducers:

1. Obtain the transducer resolution (usually from the transducer data sheet), for example 0.005 mm.
2. Determine the position units you would like for your Actual Position, for example inches, meter, mm, etc.
3. Calculate the number of position units per resolution unit, for example, using inches and a resolution of 0.005 mm: (1 in. / 25.4 mm) x (0.005 mm /Res Unit) = 0.00197 in. per Resolution unit.
4. Calculate the Position Scale with the following formula:

   Position Scale = Resolution x Position Units per Resolution Unit

5. Check which direction the axis must move to increase the counts. If you want the Actual Position to increase in the same direction, clear the Invert Feedback Polarity parameter. If you want the Actual Position to increase in the opposite direction, select the Invert Feedback Polarity parameter.
6. Calculate the Position Offset:
   - Move the axis to a point where you know what the Actual Position should be (such as 0). Call this position P_0. Record the transducer counts at that position and call that value C_0.
   - Calculate the Position Offset using the following equation:

     Position Offset = P_0 - Position Scale x C_0

Rotary Transducers:

1. Obtain the resolution per revolution (usually from the transducer data sheet), for example 1024.
2. Determine the position units you would like for your Actual Position, for example inches, degrees, meters, etc.
3. Determine the number of position units per revolution, for example, 10 in per rotation, or 360° per rev.
4. Calculate the Scale with the following formula:

   Position Scale = Counts per Rev / Resolution per Rev

5. Check which direction the axis must move to increase the counts. If you want the Actual Position to increase in the same direction, clear the Invert Feedback Polarity parameter. If you want the Actual Position to increase in the opposite direction, select the Invert Feedback Polarity parameter.
6. Calculate the Position Offset:
   - Move the axis to a point where you know what the Actual Position should be (such as 0). Call this position P_0. Record the transducer counts at that position and call that value C_0.
   - Calculate the Position Offset using the following equation:

   Position Offset = P_0 - Position Scale x C_0

6.2.6.9. SSI Transducer Fundamentals

Synchronous Serial Interface (SSI) is a widely accepted controller interface. Position data from the sensor is encoded in a binary or Gray Code format and transmitted over a high-speed serial interface. Each RMC70 MA axis module has 1 or 2 axes that can be individually configured for SSI or MDT inputs. For supported SSI options, see MA Transducer Options.

SSI Advantages

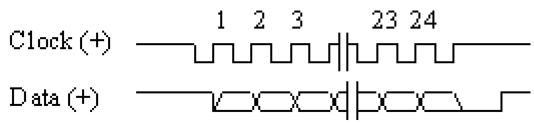
SSI transducers and absolute encoders offer the following advantages:

- High resolution. Down to 1 µm (approx. 0.00004") for linear SSI transducers.
- Noise immunity

- Cost effective data transfer (only one 5-wire cable with shield is needed)
- Transmission rate independent of data length and resolution
- Transmission over long distances
- Direct connection to the RMCs SSI axis module

Data Format

To read an SSI position, the RMC sends clock pulses to the transducer. On each rising edge of the +Clock signal, the SSI transducer places one bit of the digital position on the +Data and -Data signals. Below is shown a 24-bit SSI reading:



Note:
It is highly recommended that a Synchronized SSI transducer be selected. This ensures that the time between position samples matches the control loop time of the RMC controller. If the transducer is not synchronized, the sample time may not match and make precise speed control difficult.

The value obtained from the SSI data is put in the Raw Counts register for that axis. The Raw Counts are converted to Counts and then into an Actual Position in user-defined units.

6.3. Expansion Modules

6.3.1. Expansion Modules Overview

Expansion Modules can be added to the RMC70 series motion controller for additional functionality. These modules can be installed in the field.

The following Expansion Modules are available for the RMC70 series:

Module	Description	I/O Type	Usage
AP2	Analog Pressure	2 Analog Voltage or Current Inputs	dual-loop control (position/velocity-pressure/force)

Up to 4 expansion modules can be added to the RMC70 series motion contorller.

See also:

Physical Components Overview

RMC70 Part Numbering

Adding an Expansion Module

Adding an Expansion Module to a Controller

To add an expansion module to an RMC70 controller:

1. Disconnect power to the RMC70.
2. Remove the 4 screws on the right side of the top and bottom of the controller.
3. Plug the expansion module into the right side of the controller.
4. Install the 4 screws into the same holes they were removed from in step 2.
5. Apply power to the RMC70.
6. Open RMC70Tools and open your old project. If you do not have an old project, start a new project and do not continue this procedure.
7. If you can go online with the controller, do so. If Hardware conflict dialog opens, click **Go Online**. Save the project. Save the RMC70 settings to Flash memory by issuing the Update Flash (110) command and do not continue this procedure.
8. Right-click the controller and click **Properties**.
9. Click **Add Module**, select the module you just added, and click **OK**.
10. Click **OK**, download the settings to the controller, and go online with the controller.

6.3.2. AP2

6.3.2.1. AP2 Expansion Module

The 2-axis Analog Pressure Expansion Module (AP2) is one of the optional expansion modules available for the RMC70 series motion controller. This module provides two analog inputs with 16-bit resolution and support for pressure/force control.

The AP2 should not be confused with the A*2*, which provides analog input and no support for pressure control. The RMC70 always requires an AP2 module for pressure or force control.

Uses of the AP2 Module

The AP2 is useful for the following purposes:

- **Dual-loop Control**
    - The AP2 pressure or force inputs can be used in conjunction with the position or velocity control from an RMC7*x* Axis module toprovide the following combinations of dual-loop control:
        - **Position-Pressure**
        - **Position-Force**
        - **Velocity-Pressure** - not yet supported by the RMC7*x*.
        - **Velocity-Force** - not yet supported by the RMC7*x*.

      See dual-loop control for details.
- **Pressure/Force Control**

  The AP2 inputs, configured for pressure or force, can be used in conjunction with an output on an RMC7*x* Axis Module for solely pressure or force control. The RMC7*x* always requires an AP2 module for pressure or force control. See pressure/force control for more details.
- **Analog Reference Input**

  Reference inputs are often used as gearing or camming masters. Reference inputs cannot be used for direct control of the input axis. If the application does not require

pressure or force control, the A2 Expansion module is a better choice for adding an analog input. See Reference Axes for more details.

Input Types

The AP modules supports the following types of input:

- **Voltage:** -10 to +10 V
- **Current:** 4-20 mA

Other

AP2 Wiring

AP2 LEDs

6.3.2.2. AP2 Wiring

The AP2 expansion module can be wired to voltage or current feedback transducers.

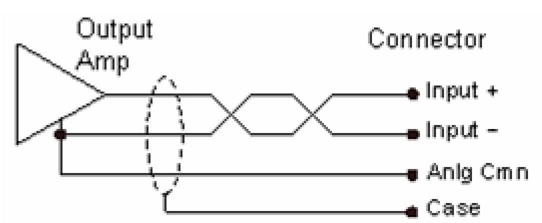
Use shielded twisted pairs for all connections to inputs and outputs. Route the transducer wiring separate from other wiring. You must provide the power supplies needed by your transducers. See Wiring Guidelines for general wiring information.

NOTE:
The example schematics do not include transducer pin numbers, color codes, or power supply requirements, since these vary between different transducers. To determine your power supply needs and connector pin-outs or cable color codes, consult your transducer manufacturers documentation.

Wiring Diagrams and Instructions

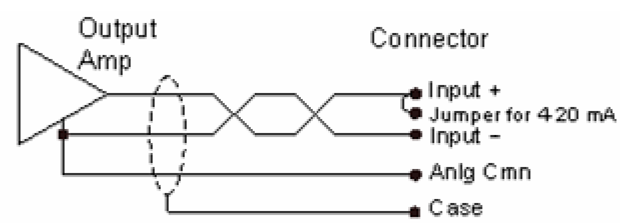
Voltage Feedback Transducers

Voltage feedback transducers can be connected directly to the **Input +** and **Input -** connections of the desired axis. The **Anlg Cmn** and **Case** pins may be shared by the axes. The following configuration is recommended:



Current Feedback Transducers

Current feedback transducers are connected in the same way as voltage transducers except that a jumper must be inserted between the **Input+** and **Jumper for 4-20 mA** pins. The label indicates where this jumper should be connected. This places a resistor internal to the RMC across the two inputs, thus converting the current to a voltage input. The following wiring diagram shows a suggested configuration:



6.3.2.3. AP2 LEDs

The AP2 module has 1 LED per axis:

Input LED

This LED represents the input and control status for the axis. This bi-color (red/green) LED has the following states:

State	Description
Steady Off	No power or not configured for use.
Steady Green	No halts have been caused by Transducer error bits (No Transducer, Transducer Overflow, and Transducer Noise) on this axis.
Steady Red	A halt has occurred on this axis because a Transducer error bit (No Transducer, Transducer Overflow, and Transducer Noise) has been set. This error can be cleared by the Clear Faults (4) command after the underlying condition has gone away.

6.3.2.4. Analog Scaling for the AP Module

To have any useful meaning, the transducer voltage from the analog pressure or force transducer must be scaled to pressure or force units. The Pressure/Force Scale, Pressure/Force Offset, and Invert Feedback Polarity parameters are used to define pressure or force units as a function of transducer voltage. This topic describes how to correctly calculate these parameters for an analog pressure transducer.

Scaling Voltage to Single-Input Pressure or Force Units

The RMC70 calculates the Actual Pressure/Force every control-loop using the following formula (may be either pressure or force):

Actual Pressure = (Voltage x Pressure Scale) + Pressure Offset

To correctly scale the axis for your particular application, use one of the methods below to find the Pressure/Force Scale, Pressure/Force Offset, and Invert Feedback Polarity parameters.

Method 1: Use Transducer Specifications

- From the transducer data sheet, find the maximum output voltage of your pressure transducer and the pressure at that voltage.
- Calculate the Pressure Scale with the following formula:

  Pressure Scale = Max Pressure / Max Voltage

  - Make sure the system has 0 pressure and then record the value of the Voltage register. Enter the negative of that value as the Pressure Offset.

Method: P_0/P_1 Calculation

The accuracy of this method depends on how accurately you can measure the pressure of the axis.

1. Physically measure two different pressures of the axis and record the value of the Voltage register at each point. (The difference between the two pressure should be as large as possible). Call the *smaller* measured pressure P_0, and its corresponding Voltage V_0. Call the *greater* measured pressure P_1, and its corresponding Voltage V_1.
2. Calculate the Pressure Scale with the following equation:

   Pressure Scale = $(P_0-P_1)/(V_0-V_1)$

   If the Pressure Scale is negative, ignore the minus sign. RMC70Tools accepts only a positive Scale.

5. If you calculated a negative Pressure Scale, select the Invert Feedback Polarity parameter. If you calculated a positive Pressure Scale, this parameter should be cleared.
6. Calculate the Pressure Offset with the following equation:

   Pressure Offset = P_0 - Pressure Scale x V_0

Scaling Voltage to Dual-Input Force Units

Dual-Input Force requires two pressure transducers. Each pressure transducer must be individually scaled. The RMC70 calculates the Actual Differential Force every control-loop using the following formulas:

Actual Force A = (Voltage x Force A Scale) + Force A Offset

Actual Force B = (Voltage x Force B Scale) + Force B Offset

Actual Differential Force = Actual Force A - Actual Force B

To correctly scale the axis for your particular application, use the method below to find the A and B Force Scale, Force Offset, and Invert Feedback Polarity parameters.

Method: Use Transducer Specifications

- From the transducer data sheet, find the maximum output voltage of your pressure transducer and the pressure at that voltage.
- Calculate the area of the A side (cap end) of the piston.
- Multiply the area of the A side with the maximum pressure of the A transducer to get the maximum A force.
- Calculate the Force A Scale with the following formula:

  Force A Scale = maximum A force / Max A Voltage
- Calculate the effective area of the B side (rod end) of the piston. This is the area of the cylinder minus the area of the rod.
- Multiply the area of the B side with the maximum pressure of the B transducer to get the maximum B force.
  - Calculate the Force B Scale with the following formula:

    Force B Scale = maximum B force / Max B Voltage
  - Make sure the system has 0 force and then record the value of the Voltage A register. Enter the negative of that value as the Force A Offset.

7. Troubleshooting

7.1. Troubleshooting Overview

> **Tip:**
> If you issued a command, but nothing happened, open the Event Log to see if the command caused any errors.

Tools

The following tools are useful for troubleshooting:

Tools	Description
Event Log	Logs every command and any command errors. Provides Error Codes for the errors that have occurred.
Plots	Plot any register in the RMC70. Track motion and see what happened at each point.
Status Bits	Provide an overview of the state of each axis.
Error Bits	Provide an overview of the errors that have occurred on the axis.

Tips

- Use the Event Log Monitor to view the commands you have issued and any Error Codes they may have caused.
- Add the Status Bits register and Error Bits register to the plot data.

Plots

I can't trigger a plot:

- Make sure the plot trigger is enabled. Use the Enable/Disable Plot Trigger (104) command.
- Make sure the plot is re-armed. Use the Rearm Plot (103) command.

Support

If you are unable to solve a problem, contact Delta's Technical Support.

7.2. Error Codes

When any error occurs in the RMC70, an error code is reported. The error codes, along with all controller events, are recorded in the Event Log. This log can be viewed in the Event Log Monitor in RMC70Tools. There are four groups of error codes, each setting a different Error Bit.

The most recent error from any of these four groups is stored in the Last Error Number status register.

Error Code Descriptions

Command Errors

These errors set the Command Error bit.

> **Note:**
> The Command Error bit is cleared when any valid command is issued.

No.	Name
1	Invalid command The command number itself is not supported by the axis it was issued to.Either that command number is reserved for future commands, or it is a defined command that is not supported by this axis type.
2	Command not implemented This error indicates that the command has not been implemented in the firmware. This error should never occur in production firmware. Please report this to Delta.
3	Invalid command parameter 0 The first command parameter had an invalid value.Refer to the commands documentation for details on valid ranges for each command parameter.
4	Invalid command parameter 1 The second command parameter had an invalid value. Refer to the commands documentation for details on valid ranges for each command parameter.
5	Invalid command parameter 2 The third command parameter had an invalid value. Refer to the commands documentation for details on valid ranges for each command parameter.
6	Invalid command parameter 3 The fourth command parameter had an invalid value. Refer to the commands documentation for details on valid ranges for each command parameter.
7	Invalid command parameter 4 The fifth command parameter had an invalid value. Refer to the commands documentation for details on valid ranges for each command parameter.
8	Invalid time calculated This error occurs when the Speed At Position (36) command is issued with starting conditions and command parameters that lead to the RMC computing a negative time to make the move in. This indicates an impossible request. See the Speed at Position (36) topic for details on conditions and parameter values that cause this error.
9	Unable to clear Halt condition

A motion command was issued, but the attempt to clear the error bits before processing the command was not successful. In order to issue a motion command, you must first clear all error conditions that cause a halt.

10 Unable to clear transducer errors

A motion command that requires transducer feedback (e.g. closed loop moves) was issued, but transducer errors that cause a halt remained on the axis.

11 Axis is not initialized

A motion command was issued, but the axis has not been initialized with the Initialize Axis (7) command.

12 Pressure/Force Limit not allowed in Direct Drive

An Enable/Disable Pressure Limit (40) command was issued to enable P/F limit, but the axis was in Direct Drive. The axis cannot be in Direct Drive while limiting Pressure/Force.

13 Unspecified error trying to start at the requested program

An internal error prevented the Start Task (90) command from executing correctly. Please report this to Delta.

14 User Programs are stopped or not loaded

The Start Task command was not taken because either no User Programs were loaded, or the User Programs were not in Run mode.

15 Target pressure must be set before enabling pressure limit

Issue the Ramp Pressure (41) to set the desired target pressure before enabling pressure with the Enable/Disable Pressure Limit (40) command.

Command Modified Errors

These errors set the Command Modified error bit.

No.	Name
81	Requested position truncated at limit

You issued a motion command with a Requested Position that was outside the Positive or Negative Overtravel Limits. The requested position was truncated at the limit or the current position, whichever is farther from the limits. Check the Positive and Negative Overtravel limits.

> **Note:**
> If you want the axis to halt instead of moving to a truncated position, make sure the Auto Stop on the Command Adjusted error bit is set to halt.

82 Requested pressure/force truncated at limit

The Ramp Pressure/Force (41) command requested a pressure/force outside the Positive or Negative Limits. The requested pressure/force was truncated at the limit or the current pressure/force, whichever is farther from the limits. Check the Overtravel limits.

> **Note:**
> If you want the axis to halt instead of moving to a truncated pressure or force,

make sure the Auto Stop on the Command Adjusted error bit is set to halt.

Configuration Errors

These errors set the Configuration Error bit.

Note:
The Configuration Error bit is cleared when any valid parameter write occurs.

No.	Name
101	Value out of range The value written was out of range for that register. The write was ignored.
102	Non-zero write to a reserved register A reserved register was written to with a non-zero value.This has no affect, but because it was unlikely that this was intentional, we record this as an error.
103	Invalid data field A bit-field in a register was incorrect. This applies to registers that are made up of several bit fields. One or more of its components were incorrect.
104	Write to a read-only register A value was written to a read-only register. The write was ignored.
105	Write to an un-implemented register A value was written to a register that does not currently support writes, but should support writes.The write is ignored. This error should never occur in production firmware. Please contact Delta technical support if this occurs.
106	The value was truncated to fit in range The value written was out of range for that register. The value was truncated to fit into range and then used. Compare with error 101.
107	The value was rounded to the nearest acceptable value The value was not an acceptable value, but was within range. It was rounded to the nearest acceptable value and then used. For example, the sample periods on a plot must be a whole number of control loop times. In this case, the value will be rounded to the nearest whole number of loop times.
108	One or more Auto Stop settings were invalid One or more of the Auto Stop settings in this register were out of range. This means that either a reserved value was used or a Auto Stop setting that is not allowed for a particular error. For example, status only cannot be selected for No Transducer.
109	This value cannot be changed while in Run mode This error indicates that an attempt was made to change the number of available Tasks or enable/disable the PreScan Table while the controller is in RUN mode. These registers can currently only be set through RMC70Tools, which will automatically put the controller temporarily in STOP mode.

Runtime Errors

These errors set the Runtime Error bit.

No.	Name
201	Command block dropped A command or set of commands that was issued to the controller was dropped because too many sets were received simultaneously. The RMC70 allows up to four command sets (up to one command per axis in each set) to be queued up before losing a command set. One command set is processed (and thus, removed from the queue) per control loop. It is possible, although unlikely, to overflow the queue by simultaneously issuing commands from multiple sources

for consecutive control loops.

202 FLASH update failed

An attempt to save the RMC settings to FLASH failed. This indicates a failure of the RMC hardware. Please contact Delta technical support if this occurs.

203 Unable to initialize the PROFIBUS sub-system

The PROFIBUS sub-system of the hardware (RMC75P only) could not be started. This indicates a failure of the RMC hardware.Please contact Delta technical support if this occurs.

204 Target Position went out of limits

The Target Position of an axis went out of limits. Notice that this does not occur because the axis is commanded to go outside of limits, as that is caught by Command Adjusted error 81, but it indicates that a move was given a low deceleration ramp that caused it to move outside the limits.

205 Unsupported FLASH command received

This is an internal error. Please contact Delta technical support if this occurs.

206 Reserved

207 Reserved

208 Drive disabled due to Watchdog timeout

This indicates an internal error occurred that caused the control outputs to be disabled. Please contact Delta technical support if this occurs.

209 Commands overwritten in the PROFIBUS command area

Commands were left in the PROFIBUS command area when either the first deferred command or a singular command was issued. All commands that had been in the command area are discarded without further processing.

210 Command overwritten in the PROFIBUS command area

A command was issued over the top of a deferred command in the PROFIBUS command area. The deferred command is discarded without further processing.

211 PROFIBUS read/write length invalid

The length field in the PROFIBUS Enhanced Data Channel had an invalid value.

212 Reserved

213 Reserved

214 Reserved

215 Reserved

216 Internal Target Generator fault

An internal target generator fault occurred. This error will always halt the axis, regardless of the Auto Stop setting, as is required because the current target generator cannot be used when faulted.

7.3. Technical Support

Delta Technical Support Contact Information

Phone: 360-254-8688

Fax: 360-254-5435

Pager: 360-699-7784 (24 hr)

Email: support@deltamotion.com

Website: http://www.deltamotion.com

RMC Return for Repair

If you need to return the RMC for repair, please contact Delta prior to shipment for an RMA number. Returned RMCs must be packaged in static protection material and have the RMA number clearly marked on the outside of the package.

Please include a short note explaining the problem!

Send the controller to:

Delta Computer Systems, Inc.

11719 NE 95th St., Suite D

Vancouver, WA 98682-2444

8. Registers

8.1. RMC70 Register Address Format

A *register* is a place in the RMC70 memory that stores data. The registers in the RMC70 are 32 bits and are either floating point or integer format. When in integer format, each bit contains individual information. See the Register Map for a complete list of the registers in the RMC70.

This topic describes the address format for the RMC70 as displayed in RMC70 Tools, and used by the PROFIBUS and DF1 communications types. For a description of the address format used by the serial Modbus RTU protocol, see the Register Format - Modbus RTU topic.

Addresses

Registers

The address format for a register is:

Fn:x

where n = File number and x = the register number as given in this topic.

Individual Bits

Some registers contain individual bits for data items. The address for a specific bit is:

Fn:x.b, where n and x are defined above, and b = bit number.

Example:

The user would like the address for the Enable Output Status Bits on axis 0. Then,

n = 8, the file number for Axis 0 Status registers,

x = 0, the number of the Status Bits register,

and

b = 7, the bit number for the Enable Output bit.

Therefore, the address is F8:0.7

Register Addresses in Integer Format

This section describes how to convert register addresses from the standard register representation to an integer.

RMC70 addresses are normally represented in the following format:

Fn:x, where n = File number, and x = Element number.

Use the following equation to convert a register address to integer format, N:

N = n * 4096 + x

Example:

Register address F8:33 is 8*4096 + 33 = 32801.

Register Address Format for Modbus RTU

The Modbus RTU protocol does not support the standard RMC70 register address format. This topic describes how to convert the RMC70 register addresses to Modbus RTU format.

Converting to Modbus RTU Address Format

The RMC70 address format is: F*file*:*element*, for example F12:8.

Use the following formula to convert RMC70 address to Modbus RTU format:

Modbus RTU Address = 2 x [(256 x *file*) + *element*] + (40001 or 400001)

Examples:

F12:3 = 2 x [(256 x 12) + 3] + 400001 = 406151

F9:56 = 2 x [(256 x 9) + 56] + 400001 = 404721

8.2. RMC70 Register Map

A *register* is a place in the RMC70 memory that stores data. The core of the RMC70 communication is the register map. It lists the addresses for all the data available in the RMC70.

The register map is divided into register files each with multiple elements (registers). Each register is represented with a letter indicating the data type (F for float, L for 32-bit integer), followed by the file number, and then a colon (:) and finally the element number. For example, F8:0 represents element 0 (the first element) in the floating point file 8.

Some protocols do not support this address format, such as Modbus/RTU. To convert the registers in this register map to Modbus RTU format, see the Register Format - Modbus RTU topic.

The following register map is a listing of all the registers in the RMC70.

File	Description
F7	Controller Info

Element	Register Name
F7:0	Product ID
F7:1	State
F7:2	CPU Model ID
F7:3	CPU Model Rev
F7:4	Axis Model ID
F7:5	Axis Model Rev
F7:6	Expansion 1 Model ID

F7:7	Expansion 1 Model Rev
F7:8	Expansion 2 Model ID
F7:9	Expansion 2 Model Rev
F7:10	Expansion 3 Model ID
F7:11	Expansion 3 Model Rev
F7:12	Expansion 4 Model ID
F7:13	Expansion 4 Model Rev
F7:14	Serial Number (Long Integer)
F7:15	Firmware Rev
F7:16	Firmware Beta Code
F7:17	Firmware Configuration ID
F7:18	Firmware Year and Month
F7:19	Firmware Day and Time
F7:20	FPGA Rev
F7:21	FPGA Configuration ID
F7:22	FLASH Rev
F7:23	Required RMC70Tools Ver
F7:24	Suggested RMC70Tools Ver
F7:25	Loader Rev
F7:26	Loader Year and Month
F7:27	Loader Day and Time

F8 Control Axis 0 Status Registers

Element	**Register Name**
Registers 0 - 4: Common	
	All Axes
F8:0	Status Bits
F8:1	Error Bits
F8:2	Last Error Number
F8:4	Read Response

Registers 8 - 15: Primary Input

	Position Axes	**Velocity Axes**	**Single Input Pressure/Force Axes**	**Dual Input Force Axes**
F8:8	Actual Position		Actual Pressure/Force	Actual Differential Force
F8:9	Actual Velocity	Actual	Actual Pressure/Force	Actual Differential

		Velocity	Rate	Force Rate
F8:10	Actual Acceleration	Actual Acceleration		Actual Force A
F8:11	Counts/Voltage	Voltage	Voltage	Voltage A
F8:12	Raw Counts	Raw Counts	Raw Counts	Raw Counts A
F8:13				Actual Force B
F8:14				Voltage B
F8:15				Raw Counts B

Registers 18 - 20: Home/Latch

	Quadrature Axes
F8:18	H/L Status Bits
F8:19	Latched Position 0
F8:20	Latched Position 1

Registers 23 - 30: Secondary Input

	Single Input Pressure/Force Axes	**Dual Input Force Axes**
F8:23	Actual Pressure/Force	Actual Differential Force
F8:24	Actual Pressure/Force Rate	Actual Differential Force Rate
F8:25		Actual Force A
F8:26	Voltage	Voltage A
F8:27	Raw Counts	Raw Counts A
F8:28		Actual Force B
F8:29		Voltage B
F8:30		Raw Counts B

Register 33: Output

	Analog Axes
F8:33	Control Output

Registers 35-43: Primary Control

	Position Axes	Velocity Axes	Single Input Pressure / Force Axes
F8:35	Position Error		Pressure/Force Error
F8:36	Velocity Error	Velocity Error	
F8:37	Proportional Term	Velocity Integral term	Pressure/Force Proportional Term
F8:38	Integral Term		Pressure/Force Integral Term
F8:39	Differential Term	Velocity Proportional Term	Pressure/Force Differential Term
F8:40			Pressure/Force Feed Forward Term
F8:41	Velocity Feed Forward Term	Velocity Feed Forward Term	Pressure/Force Rate Feed Forward Term
F8:42	Acceleration Feed Forward Term	Acceleration Feed Forward Term	
F8:43	PFID Output	PFID Output	PFID Output

Registers 44-50: Secondary Control

	Servo Pressure / Force Axes
F8:44	Pressure/Force Error
F8:45	Reserved
F8:46	Pressure/Force Proportional Term
F8:47	Reserved
F8:48	Pressure/Force Differential Term
F8:49	Pressure/Force Feed Forward Term

F8:50	Pressure/Force Rate Feed Forward Term

Registers 53-57: Primary Target

	Position Axes
F8:53	Target Position
F8:54	Target Velocity
F8:55	Target Acceleration
F8:56	Command Position
F8:57	Command Velocity

Registers 60-61: Secondary Target

	Pressure/Force Axes
F8:60	Target Pressure/Force
F8:61	Command Pressure/Force

F9 Control/Reference Axis 1 Status Registers - Identical to file F8

F10 Reference Axis 2 Status Registers - Identical to file F8

F11 Reference Axis 3 Status Registers - Identical to file F8

F12 Control Axis 0 Parameter Registers

The Parameter Registers are axis-dependent.

Element	Register Name			
Registers 0 - 13: Primary Feedback				
	Position Axes	**Velocity Axes**	**Single-Input Pressure/Force Axes**	**Dual-Input Force Axes**
F12:0	Position Scale	Position Scale	Pressure/Force Scale	Force A Scale
F12:1	Position Offset	Velocity Offset	Pressure/Force Offset	Force A Offset
F12:2	Actual Position Filter	Velocity Deadband		Force B Scale
F12:3	Actual Velocity	Actual Velocity		Force B Offset

	Filter	Filter		
F12:4	Actual Acceleration Filter	Actual Acceleration Filter	Actual Pressure/Force Filter	Actual Pressure/Force Filter
F12:5	Stop Threshold	Stop Threshold	Actual Pressure/Force Rate Filter	Actual Pressure/Force Rate Filter
F12:6	Noise Error Rate	Noise Error Rate	Noise Error Rate	Noise Error Rate
F12:9.0	Linear/Rotary	Linear/Rotary	Linear/Rotary	Linear/Rotary
F12:9.1	Absolute/Incremental	Absolute/Incremental	Absolute/Incremental	Absolute/Incremental
F12:9.2				
F12:9.3				

	MDT/SSI Transducer	**Analog Transducer**
F12:10	0-2 MDT Type 8 MDT/SSI 9 SSI Format (Binary/Gray) 10-11 SSI Clock Rate 12-17 SSI Data Bits	0-2 Input Range
F12:11	Count Offset	
F12:12	Position Unwind	

Registers 18-28: Secondary Feedback

	Single Input Pressure/Force Axes	**Dual Input Force Axes**
F12:18	Pressure/Force Scale	Force A Scale
F12:19	Pressure/Force Offset	Force A Offset
F12:20		Force B Scale
F12:21		Force B Offset
F12:22	Actual Pressure/Force Filter	Actual Pressure/Force Filter
F12:23	Actual Pressure/Force Rate Filter	Actual Pressure/Force Rate Filter

F12:24	Noise Error Rate	Noise Error Rate
F12:27.0	Linear/Rotary	Linear/Rotary
F12:27.1	Absolute/Incremental	Absolute/Incremental

	Analog Transducer
F12:28.0-2	Input Range

Registers 32-34: Output

	Analog
F12:32	Output Limit
F12:33	Output Bias
F12:34.0	Invert Output Polarity
F12:34.1	Reserved
F12:34.2	Enable Output Behavior
F12:34.3	Fault Input Polarity
F12:34.4	Limits Polarity

Registers 38-42: Final Output Stage

	Servo
F12:38	Output Scale
F12:39	Output Filter
F12:40	Directional Gain Ratio
F12:41	Deadband Tolerance
F12:42	Output Deadband

Registers 56-67: Primary Control

	Servo Position Axes	**Servo Velocity Axes**	**Servo Pressure/Force Axes**
F12:56	In Position Tolerance		At Pressure/Force Tolerance

F12:57	Position Error Tolerance		Pressure/Force Error Tolerance
F12:58	At Velocity Tolerance	At Velocity Tolerance	
F12:59	Velocity Error Tolerance	Velocity Error Tolerance	
F12:60. 0-3	Integrator Mode	Integrator Mode	Integrator Mode
F12:61	Position Proportional Gain	Velocity Integral Gain	Pressure/Force Proportional Gain
F12:62	Position Integral Gain		Pressure/Force Integral Gain
F12:63	Position Differential Gain	Velocity Proportional Gain	Pressure/Force Differential Gain
F12:64			Pressure/Force Feed Forward
F12:65	Velocity Feed Forward	Velocity Feed Forward	Pressure/Force Rate Feed Forward
F12:66	Acceleration Feed Forward	Acceleration Feed Forward	
F12:67	Integrator Limit	Integrator Limit	Pressure/Force Integrator Limit

Registers 76-85: Secondary Control

	Servo Pressure/Force
F12:76	At Pressure/Force Tolerance
F12:77	Pressure/Force Error Tolerance
F12:80. 5-6	Pressure/Force Orientation
F12:81	Pressure/Force Proportional Gain
F12:82	Pressure/Force Integral Gain
F12:83	Pressure/Force Differential Gain
F12:84	Pressure/Force

	Feed Forward
F12:85	Pressure/Force Rate Feed Forward

Registers 92-93: Position Target

	Position
F12:92	Positive Travel Limit
F12:93	Negative Travel Limit

Registers 100-101: Pressure/Force Target

	Position
F12:100	Positive Pressure/Force Limit
F12:101	Negative Pressure/Force Limit

Registers 106-113: Halts

	All Axes
F12:106 .0-2	Auto Stop Following Error
F12:106 .3-5	Auto Stop Integrator Saturated
F12:106 .6-8	Auto Stop Output Saturated
F12:106 .9-11	Auto Stop Fault Input
F12:106 .12-14	Auto Stop Positive Limit Input
F12:106 .15-17	Auto Stop Negative Limit Input
F12:106 .18-20	Auto Stop No Transducer
F12:106 .21-23	Auto Stop Transducer Overflow
F12:107	Auto Stop

.0-2	Noise Error
F12:107 .3-5	Auto Stop Positive Overtravel
F12:107 .6-8	Auto Stop Negative Overtravel
F12:107 .9-11	Auto Stop Command Error
F12:107 .12-14	Auto Stop Command Modified
F12:107 .15-17	Auto Stop Configuration Error
F12:107 .18-20	Auto Stop Runtime Error
F12:108 .6-8	Auto Stop No Transducer (Secondary)
F12:108 .9-11	Auto Stop Noise Error (Secondary)
F12:108 .12-14	Auto Stop Following Error (Secondary)
F12:110	Closed Loop Halt Deceleration
F12:111	Open Loop Halt Ramp
F12:112	Halt Group Number

Registers 116-119: Simulation

	All Axes
F12:116 .0	Simulate Mode
F12:117	System Gain (Simulate Mode)
F12:118	Natural Frequency (Simulate Mode)
F12:119	Damping Factor

(Simulate Mode)

F13 Control/Reference Axis 1 Parameter Registers - Identical to file F12

F14 Reference Axis 2 Parameter Registers - Identical to file F12

F15 Reference Axis 3 Parameter Registers - Identical to file F12

F16 Command Area

Element	Register Name
F16:0	Axis 0 Command
F16:1	Axis 0 Parameter 1
F16:2	Axis 0 Parameter 2
F16:3	Axis 0 Parameter 3
F16:4	Axis 0 Parameter 4
F16:5	Axis 0 Parameter 5
F16:6-11	Axis 1 Command Registers
F16:12-17	Axis 2 Command Registers
F16:18-23	Axis 3 Command Registers

F17 Indirect Data Map

Element	Register Name
F17:0-31	Indirect Data Map

Note:
The Indirect Data Map (registers F17:0-31) contains the *addresses* of the referenced registers, not the *data*. Do not read these registers when accessing the Indirect Data via a PLC. See the Indirect Data registers (F18:0-31).

F18 Indirect Data

Element	Register Name
F18:0-31	Indirect Data Values

Note:
These registers (F18:0-31) are the registers that should be accessed at runtime by a PLC when reading the Indirect Data.

F19 Axis Assignment - Do not attempt to change these registers. They are used by the RMC70 for assigning axes to hardware.

F20 Controller Parameters/Status

Element	Register Name	Notes
F20:0	Loop Time, Set	Read-Only
F20:1	Loop Time, Requested	Do not use this register!
F20:2	Loop Time, Length of Last Used	Read-Only

F20:3	Loop Time, Length of Maximum Used	R/W, but can only be set to 0.
F20:4	Number of Tasks Allocated (0-4)	default = 2
F20:5	PreScan Table Enabled? (0=false, 1= true)	default = 1
F20:6	Stop All Tasks on Any Axis Halt? (0=false, 1= true)	default = 1
F20:7	Controller Status Register	
F20:7.0	RUN/PROGRAM (0=PROGAM, 1= RUN)	
F20:7.1	Axes Initialized	
F20:7.2	First Scan	
F20:8-31	Reserved	

F21 Communication Configuration

These registers are module-dependent.

Element	RMC75S	RMC75P	Notes
F21:0	RS-232 Monitor Baud Rate	RS-232 Monitor Baud Rate	Read-Only
F21:1	RS-232 Monitor Configuration Bits	RS-232 Monitor Configuration Bits	Read-Only
F21:2	RS-232 Monitor Protocol	RS-232 Monitor Protocol	Read-Only
F21:3	RS-232 Monitor Address	RS-232 Monitor Address	Read-Only
F21:4	Comm. Serial Baud Rate	PROFIBUS Slave Address	
F21:5.0-1	Comm. Serial Data Bits		
F21:5.2	Comm. Serial Stop Bits		
F21:5.3-4	Comm. Serial Parity		
F21:5.5-6	Comm. Serial Line Driver		
F21:5.7	Comm. Serial Checksum Type		
F21:5.8-9	Comm. Serial Data Type		
F21:6	Comm. Serial Protocol		
F21:7	Comm. Serial Address		

F24 Task Status

Element	Register Name
F24:0-15	**Task 0 Status**
F24:0	State
F24:0.0	Running?
F24:0.1	Allocated?
F24:1	Current Step

F24:1.0-11	Step Number
F24:1.12-23	Program Number
F24:2	Current Axis (0..3)
F24:3-15	Reserved

F24:16-31	Task 1 Status (same as Task 0 Status)
F24:32-47	Task 2 Status (same as Task 0 Status)
F24:48-63	Task 3 Status (same as Task 0 Status)

F31	Plot Schema
F32	Plot 0 Configuration
F33	Plot 1 Configuration
F34	Plot 2 Configuration
F35	Plot 3 Configuration
F36	Plot 4 Configuration
F37	Plot 5 Configuration
F38	Plot 6 Configuration
F39	Plot 7 Configuration
F40	Plot 0 Status
F41	Plot 1 Status
F42	Plot 2 Status
F43	Plot 3 Status
F44	Plot 4 Status
F45	Plot 5 Status
F46	Plot 6 Status
F47	Plot 7 Status
F56	Variables - Current Values

Element	Register Name

F56:0-63	Variables - Current Values

F64 Variables - Initial Values

Element	Register Name
F64:0-63	Variables - Initial Values

8.3. Status Registers

8.3.1. Axis Status Registers Overview

The Status Registers provide information on the status of each axis. These registers are not editable; they are read-only. Each Register is a 32-bit word. The Status Registers are axis dependant. Each group listed below contains Status Registers for the specified axes. For details on the addressing format for the registers, see the Register Address Format topic.

For a list of the Status Registers, see the Register Map topic.

8.3.2. Common

8.3.2.1. Error Bits Register

Type:	Axis Status Register
Address:	F*n*:1, where *n* = 8 + the axis number
Format:	Integer

Description

The Error Bits register is an Axis Status Register. It is a collection of bits that provide a summary of the errors on the axis. An error bit is set when certain conditions occur. Unless otherwise noted, these bits do not clear unless a Clear Faults command or any other motion command is issued *and* the underlying error condition has gone away.

When a bit is set, a halt will occur if the Auto Stops are configured to do so and the axis is not in the Direct Drive state.

See the Register Address Format topic for details on how to address a specific bit.

Bits

Bit	Register Name
0	Following Error If the axis is in closed loop *position* control, this bit is set when the Position Error exceeds the Position Error Tolerance parameter. If the axis is in closed loop *velocity* control, this bit is set when the Velocity Error exceeds the Velocity Error Tolerance parameter.
1	Integrator Saturated

If the axis is in closed loop position or velocity control, this bit is set when the integrator is saturated. The integrator is saturated when the Integral Output Term reaches the Integrator Limit parameter.

2 Output Saturated

If the Control Output is greater than the Output Limit parameter, then the Control Output is saturated and this bit is set. This bit is not set when the Control Output is limited at the DAC output stage. It is just silently truncated. Truncation at this stage results from applying the Output Bias and/or Output Polarity.

3 Fault Input

This bit is set when the Fault input for the axis goes active. Notice that there is also a Fault Input State status bit that is not latched as this error bit is.

4 Positive Limit Input

This bit is set when the Positive Limit input is active.

5 Negative Limit Input

This bit is set when the Negative Limit input is active.

6 No Transducer

This bit is set when The RMC70 detects that no transducer is connected.

7 Transducer Overflow

This bit is set when the RMC70 detects an overflow on the input from the transducer.

8 Noise Error

If the input rate of change on the transducer input exceeds the Noise Error Rate for a period of 10 ms, then this error bit is set. During noise less than 10 msec, the Actual Position is estimated to allow recovery from electrical noise.

9 Positive Overtravel

This bit is set when the axis travels beyond the Positive Travel Limit and is not in the Direct Drive state.

> **Note:**
> This bit turns on only when exceeding the limit or attempting to go further past the limit. It will turn off if a command is issued that keeps the axis in the same place or moves it toward the limit.

10 Negative Overtravel

This bit is set when the axis travels beyond the Negative Travel Limit and is not in the Direct Drive state.

> **Note:**
> This bit turns on only when exceeding the limit or attempting to go further past the limit. It will turn off if a command is issued that keeps the axis in the same place or moves it toward the limit.

11 Command Error

When an invalid command or command with invalid parameters is issued, this bit is set and an error number is stored in the Last Error Number status register. Depending on the Event Log filter settings, this error may also be logged to the Event Log. See the Error Codes topic for a list of errors that cause this bit to be set. This bit is cleared when any valid command is issued.

12 Command Modified

This bit is set when a command has been modified. If a command

contains an invalid Command Parameter, it may be modified. An error number will be stored in the Last Error Number status register. Depending on the Event Log filter settings, this error may also be logged to the Event Log. See the Error Codes topic for a list of errors that cause this bit to be set. This bit is cleared when any valid command is issued.

13 Configuration Error

When an attempt is made to write an invalid value to a parameter, this bit is set and an error number is stored in the Last Error Number status register. Depending on the Event Log filter settings, this error may also be logged to the Event Log. See the Error Codes topic for a list of errors that cause this bit to be set. This bit is cleared when any valid parameter write occurs.

14 Runtime Error

When an unexpected condition occurs that does not have its own error bit, this bit is set and an error number is stored in the Last Error Number status register. Depending on the Event Log filter settings, this error may also be logged to the Event Log. See the Error Codes topic for a list of errors that cause this bit to be set.

15 Reserved

This bit has no current use.

16 Reserved

This bit has no current use.

17 Reserved

This bit has no current use.

18 Pressure/Force No Transducer

This error bit is set in the same way that the No Transducer error bit is set, except this bit is set based on the secondary analog pressure/force input.

19 Pressure/Force Transducer Overflow

This error bit is set in the same way that the Transducer Overflow error bit is set, except this bit is set based on the secondary analog pressure/force input.

20 Pressure/Force Noise Error

This error bit is set in the same way that the Noise Error bit is set, except this bit is set based on the secondary analog pressure/force input.

21 Pressure/Force Following Error

This error bit is set in the same way that the Following Error bit is set, except this bit is based on the pressure/force on an axis with pressure/force limit.

8.3.2.2. Last Error Number Register

Type:	Axis Status Register
Address:	F*n*:2, where *n* = 8 + the axis number
Format:	Floating Point

Description

This status register stores the number of the last error that occurred.

8.3.2.3. Read Response

Type:	Axis Status Register
Address:	F*n*:4, where *n* = 8 + the axis number
Format:	Floating Point

Description

This status register stores the value returned by the Read Register (111) command.

9.

9.1.1.1. Status Bits Register

Type:	Axis Status Register
Address:	F*n*:0, where *n* = 8 + the axis number
Format:	Integer - See Format Details below

Description

The Status Bits register is a collection of bits that provide a summary of the state of the axis. See the Register Address Format topic for details on how to address each specific bit.

Status Bits

#	Register Name
0	In Position This bit is set when the axis position is at the Command Position. This is defined as when the absolute difference between the Actual Position and the Command Position is less than the In Position Tolerance parameter. The axis must be in closed-loop control and the Primary Target Generator Done bit must be set, which means the Target Position is at the Command Position.
1	At Velocity This bit is valid only for velocity axes. It is set when the axis velocity is at the Command Velocity. This is defined as when the absolute difference between the Actual Velocity and the Command Velocity is less than the At Velocity Tolerance parameter. The Primary Target Generator Done bit must be set for this bit to be set.
2	Open Loop This bit is set when the axis is in Open Loop control or in the Direct Drive state.
3	Fault Input This bit is set when the fault input on the axis indicates a fault condition.
4	Positive Limit Input This bit is set when the Positive Limit Input indicates it is on.
5	Negative Limit Input This bit is set when the Negative Limit Input indicates it is on.
6	Stopped This bit is set when the axis is stopped. This is defined as when the Target Velocity is zero and the Actual Velocity is less than the Stop Threshold. The axis must be in closed-loop control.
7	Input Estimated This bit indicates that the input from the transducer is currently being estimated. If the rate of change on the transducer input exceeds the Noise Error Rate for a period of less than 10 ms, the Actual Position (or Actual Velocity) is estimated to allow recovery from electrical noise. If the noise lasts longer than 10 msec, the Actual Position (or Actual Velocity) is estimated and the Noise Error bit is set.

8 Enable Output

This bit is set when the Enable Output is on.

9 Primary Target Generator Done

This bit is set when the Primary Target Generator has completed its course, that is, when a motion command has been completed. If the motion move is interrupted, e.g. due to a halt, the done bit will not be set because the last commanded motion was not completed.

10 Primary Target Generator State A

The Target Generator State A and State B bits indicate the current state of the Primary Target Generator. For each motion type, the following table describes in general the state indicated by the combinations of A and B. These bits are only valid for motion commands.

> **Note:**
> These states may differ for some motion commands. See the command for specific details.

A	B	Open Loop Commands	Point-to-Point Commands	Quick Moves	Halts and Stops
0	0	Constant Control Output at 0 V	Done	Done	Done
0	1	Ramping Control Output away from 0 V	Accelerating	Ramping Control Output in Open Loop	Reserved
1	0	Constant Control Output at non-zero V	Constant Velocity	Constant Control Output at Requested Output	Reserved
1	1	Ramping Control Output toward 0 V	Decelerating	Decelerating in Closed Loop	Decelerating or Ramping Down the Control Output

11 Primary Target Generator State B

See the Target Generator State A bit.

12 Secondary Target Generator Done

This bit is set when the Secondary Target Generator has completed its course. This bit is not currently used.

13 Initialized

This bit is set when the Initialize Axis (7) command is issued to the axis. It remains set until power the RMC70 is reset. Motion commands are not allowed on the axis until this bit has been set.

14 External Halt

This bit is set when External Halt has occurred. An External Halt can only be triggered by an Auto Stop.

15 Halted

This bit is set when one of the following halts occurs: Open Loop Halt, Open Loop Halt with Disable Drive, or Closed Loop Halt.

These halts can then be triggered in two ways:

- Via Auto Stops
- Via explicit commands - see each halt topic for details.

16 Reserved

17 Pressure/Force Limit Enabled

This bit is set when Pressure Limit mode is enabled on the axis. When this bit is set, pressure or force will be limited to the Target Pressure/Force. Pressure Limit mode can only be enabled or disabled with the Enable/Disable Pressure Limit (40) command.

18 Pressure/Force Limited

This bit is set when pressure or force on the axis is being limited. This is defined as when at least one of the gain terms in order to limit the pressure or force. See the Pressure Limit topic for details on pressure control.

19 At Pressure/Force

This bit is set when the Actual Pressure or Force is at the Command Pressure or Force. This is defined as when the Pressure/Force Error is less than the At Pressure/Force Tolerance. The axis must be in closed-loop pressure control and the Secondary Target Generator Done bit must be set.

20 Pressure/Force Input Estimated

This bit indicates that the pressure/force input is currently being estimated.

21 Pressure/Force Target Generator Done

This bit is set when the Pressure/Force Target Generator has completed its course, that is, when a pressure or force ramp has been completed. If the ramp is interrupted, e.g. due to a halt, the done bit will not be set because the ramp was not completed.

22 Pressure/Force Target Generator State A

This bit is set to indicate the current state of the Pressure/Force Target Generator.

23 Pressure/Force Target Generator State B

This bit is set to indicate the current state of the Pressure/Force Target Generator.

9.1.2. Input

9.1.2.1. Actual Acceleration

Type:	Axis Status Register
Address:	F*n*:10, where *n* = 8 + the axis number
Format:	Floating Point
Units:	pu/sec^2

Description

This is the actual acceleration of the axis. It is updated every control loop. This value is obtained differently depending on the axis type:

- **Position Feedback Axes**
  The Actual Acceleration is the calculated rate of change in the Counts status register. It is calculated as follows:

Actual Position = [(Counts(n) - Counts(n-1)) - (Counts(n-1) - Counts(n-2))] / Control Loop Time.

where n is the control loop.

- **Velocity Feedback Axes**
  The Actual Acceleration is calculated from the change in the Counts status register. It is calculated as follows:

  Actual Position = Counts x Velocity Scale

  Actual Position = (Counts(n) - Counts(n-1)) x Velocity Scale / Control Loop Time

  where n is the control loop.

If the Actual Acceleration is noisy, you can filter it. By default the Actual Acceleration is filtered to account for quantizing errors. See the Actual Acceleration Filter topic for details.

9.1.2.2. Actual Differential Force

Type:	Axis Status Register
Address:	F*n*:8, F*n*:23,where *n* = 8 + the axis number
Format:	Floating Point
Units:	F

Description

The Actual Differential Force register holds the measured differential force of the axis at any moment. This force is updated every control loop.

Differential Force is normally used in hydraulic cylinder applications. The measured pressure on each side of the cylinder is multiplied by the respective area of the piston to obtain the forces applied to piston. The difference of the absolute value of these two forces is the Differential Force applied to the cylinder rod.

To achieve differential force control, two pressure transducers must be connected to the cylinder; one on the A port (cap end) and another on the B port (rod end). Notice that it is not possible to calculate resultant force on the rod with only one pressure transducer because the pressure on the other side of the cylinder is unknown. Applications with one pressure transducer cannot use force control, but can use pressure control.

9.1.2.3. Actual Differential Force Rate

Type:	Axis Status Register
Address:	F*n*:9, F*n*:24,where *n* = 8 + the axis number
Format:	Floating Point
Units:	F

Description

The Actual Differential Force Rate register holds the measured rate of change of the axis differential force at any moment. This rate is updated every control loop.

Differential Force is normally used in hydraulic cylinder applications. The measured pressure on each side of the cylinder is multiplied by the respective area of the piston to obtain the forces applied to piston. The difference of the absolute value of these two forces is the Differential Force applied to the cylinder rod.

To achieve differential force control, two pressure transducers must be connected to the cylinder; one on the A port (cap end) and another on the B port (rod end). Notice that it is not possible to calculate resultant force on the rod with only one pressure transducer because the pressure on the other side of the cylinder is unknown. Applications with one pressure transducer cannot use force control, but may use single-input pressure control.

9.1.2.4. Actual Force A, Actual Force B

Type:	Axis Status Register
Address:	Actual Force A: F*n*:10, F*n*:25, where *n* = 8 + the axis number
	Actual Force B: F*n*:13, F*n*:28, where *n* = 8 + the axis number
Format:	Floating Point
Units:	F

Description

The Actual Force A register holds the analog transducer reading after conversion from Counts to force units.

9.1.2.5. Actual Position

Type:	Axis Status Register
Address:	F*n*:8, where *n* = 8 + the axis number
Format:	Floating Point or Special - see below
Units:	pu

Description

The Actual Position is the measured position of the axis at any moment, updated every control loop. This status register is valid only on position control axes. The Actual Position is calculated from the transducer as follows:

Actual Position = (Counts x Position Scale) + Position Offset

To properly scale the counts to position-units, see the Position Scale topic.

If the Actual Position is noisy, you can filter it. See the Actual Position Filter topic for details.

9.1.2.6. Actual Pressure/Force

Type:	Axis Status Register
Address:	F*n*:8, F*n*:23, where *n* = 8 + the axis number
Format:	Floating Point
Units:	Pr or F

Description

The Actual Pressure or Actual Force register holds the measured pressure or force of the axis at any moment. This value is updated every control loop.

Pressure requires a pressure transducer, and force requires a force transducer (strain gauge). Notice that force cannot be measured with only one pressure transducer in a hydraulic system. To measure force, two pressure transducers are required. The

measured value is then a differential force. See the differential force topic for more details.

9.1.2.7. Actual Pressure/Force Rate

Type:	Axis Status Register
Address:	F*n*:9, F*n*:24, where *n* = 8 + the axis number
Format:	Floating Point
Units:	Pr/s or F/s

Description

The Actual Pressure Rate or Actual Force Rate register holds the calculated rate of change in pressure or force of the axis at any moment. This value is updated every control loop.

9.1.2.8. Actual Velocity

Type:	Axis Status Register
Address:	F*n*:9, where *n* = 8 + the axis number
Format:	Floating Point
Units:	pu/sec

Description

This is the velocity of the axis. It is updated every control loop. This value is obtained differently depending on the axis type:

- **Position Feedback Axes**
  The Actual Velocity is calculated from the change in the Counts status register. It is calculated as follows:

  Actual Position = (Counts(n) - Counts(n-1)) x Position Scale / Control Loop Time

  where n is the control loop.

- **Velocity Feedback Axes**
  The Actual Velocity is the velocity from the transducer. It is calculated as follows:

  Actual Position = Counts x Velocity Scale

If the Actual Velocity is noisy, you can filter it. By default the Actual Velocity is filtered to account for quantizing errors. See the Actual Velocity Filter topic for details.

9.1.2.9. Counts

Type:	Axis Status Register
Address:	F*n*:11, F*n*:26, where *n* = 8 + the axis number
Format:	Floating Point
Units:	counts

Description

Counts are the feedback from the transducer. The Counts are derived from the raw counts register, which is the value obtained from the counter on the input. These counts are primarily for the user to set up and troubleshoot scaling.

The Counts register is for axes with MDT, SSI, or Quadrature transducers. Axes with analog transducers do not have a Counts register; they have a Volts register.

9.1.2.10. Volts

Type:	Axis Status Register
Address:	F*n*:11, F*n*:26, where *n* = 8 + the axis number
Format:	Floating Point
Units:	Volts

Description

The Volts register is the voltage feedback from an analog transducer. It is derived from the raw counts register, which is the value from the 16-bit analog-to-digital converter. The Voltage register is primarily for the user to set up and troubleshoot scaling.

The Volts register is for axes with analog transducers. Axes with MDT, SSI, or Quadrature transducers do not have a Voltage register; they have a Counts register.

9.1.2.11. Counts A, Counts B

Type:	Axis Status Registers
Address:	Counts A: F*n*:12, F*n*:27, where *n* = 8 + the axis number Counts B: F*n*:15, F*n*:30, where *n* = 8 + the axis number
Format:	Floating Point
Units:	counts

Description

The Counts A and Counts B registers are identical to the Counts register, except that they are specific to axes with differential force. The Counts A register holds the counts for the A pressure transducer, and the Counts A register holds the counts for the A pressure transducer.

See the Counts topic for details.

9.1.2.12. Raw Counts

Type:	Axis Status Register
Address:	F*n*:12, F*n*:27, where *n* = 8 + the axis number
Format:	Floating Point or Special - see below
Units:	raw-counts

Description

The Raw Counts are the value read directly from the transducer. They are used to derive the Counts or Volts, and can be plotted for use in troubleshooting and debugging. The Scale and Offset parameters then convert Counts to user-defined units. For certain transducer types, the Counts and Raw Counts are identical.

The Raw Counts are determined in the following manner for each transducer type:

- **MDT**
  The number of MDT clock counter cycles from the start pulse to the stop pulse (or from the beginning to the end of the PWM response). This is identical to the Counts.
- **SSI**
  The value clocked in from the transducer. This is identical to the Counts.
- **Analog**
  The digital value read from the A-to-D converter. This value is then converted to Volts.
- **Quadrature**
  The value of the hardware accumulator. This is identical to the Counts.

9.1.2.13. Raw Counts A, Raw Counts B

Type:	Axis Status Registers
Address:	Raw Counts A: F*n*:12, F*n*:15, where *n* = 8 + the axis number Raw Counts B: F*n*:27, F*n*:30, where *n* = 8 + the axis number
Format:	Floating Point
Units:	raw-counts

Description

The Raw Counts A and Raw Counts B registers are identical to the Raw Counts register, except that they are specific to axes with dual-input force. The Raw Counts A register holds the raw counts for the pressure A transducer, and the Raw Counts B register holds the raw counts for the pressure B transducer.

See the Raw Counts topic for details.

9.1.3. Output

9.1.3.1. Control Output

Type:	Axis Status Register
Address:	F*n*:33, where *n* = 8 + the axis number
Format:	Floating Point
Units:	V

Description

The Control Output register holds the actual output on the Control Output pins in Volts.

The Control Output has the Output Limit and Bias taken into account, but not the Output Polarity and DAC scaling. For closed loop, it also has the Linearization Table, Feed Forward Ratio, and Output Filter taken into account.

9.1.4. Primary Control

9.1.4.1. Acceleration Feed Forward Term

Type:	Axis Status Register

Address:	F*n*:42, where *n* = 8 + the axis number
Format:	Floating Point
Units:	% of maximum control output

Description

The Acceleration Feed Forward Term is the portion of the PFID Output contributed by the Acceleration Feed Forward Gain.

See the Acceleration Feed Forward topic for more details.

9.1.4.2. Differential Output Term

Type:	Axis Status Register
Address:	F*n*:39, where *n* = 8 + the axis number
Format:	Floating Point
Units:	% of maximum control output

Description

The Differential Output Term is the portion of the PFID Output contributed by the Differential Gain.

See the Position Differential Gain topic for more details.

9.1.4.3. Integral Output Term

Type:	Axis Status Register
Address:	F*n*:38, where *n* = 8 + the axis number
Format:	Floating Point
Units:	% of maximum control output

Description

The Integral Output Term is the portion of the PFID Output contributed by the Integral Gain.

See the Integral Gain topic for more details.

9.1.4.4. PFID Output

Type:	Axis Status Register
Address:	F*n*:45, where *n* = 8 + the axis number
Format:	Floating Point or Special - see below
Units:	% of maximum control output

Description

The PFID Output is the sum of all the PFID terms before it goes to the pre-output stage.

9.1.4.5. Position Error

Type:	Axis Status Register
Address:	F*n*:35, where *n* = 8 + the axis number
Format:	Floating Point
Units:	pu

Description

The Position Error is the difference between the Target Position and Actual Position. If this value exceeds the Position Error Tolerance while in closed loop control on a position axis, the Following Error bit will be set.

9.1.4.6. Proportional Output Term

Type:	Axis Status Register
Address:	F*n*:37, where *n* = 8 + the axis number
Format:	Floating Point
Units:	% of maximum contol output

Description

The Proportional Output Term is the portion of control output contributed by the Proportional Gain.

See Proportional Gain for more details.

9.1.4.7. Velocity Error

Type:	Axis Status Register
Address:	F*n*:36, where *n* = 8 + the axis number
Format:	Floating Point
Units:	pu/s

Description

The Velocity Error is the difference between the Target Velocity and Actual Velocity. If this value exceeds the Velocity Error Tolerance while in closed loop control on a velocity axis, the Following Error bit will be set. If the Following Error bit is set, a Halt will occur if the Auto Stops are configured to do so and the axis is not in the Direct Drive state.

9.1.4.8. Velocity Feed Forward Term

Type:	Axis Status Register
Address:	F*n*:41, where *n* = 8 + the axis number
Format:	Floating Point or Special - see below
Units:	% of maximum control output

Description

The Velocity Feed Forward Term is the portion of control output contributed by the Velocity Feed Forward.

See the Velocity Feed Forward topic for more details.

9.1.4.9. Jerk Feed Forward Term

Type:	Axis Status Register
Address:	F*n*:43, where *n* = 8 + the axis number
Format:	Floating Point
Units:	% of maximum control output

Description

The Jerk Feed Forward Term is the portion of the PFID Output contributed by the Jerk Feed Forward Gain.

See the Jerk Feed Forward topic for more details.

9.1.5. Primary Target

9.1.5.1. Command Position

Type:	Axis Status Register
Address:	F*n*:56, where *n* = 8 + the axis number
Format:	Floating Point
Units:	pu

Description

The Command Position is the requested position with travel limits applied. If the requested position is outside the Positive or Negative Travel Limit, the Command Position will be set to the value of the limit, and the axis will go only to the limit. The Command Position is updated when any motion command is issued with a Requested Position command parameter.

Why Bother?

If the Command Position is not the same as the requested position then one of two things has happened:

1. A program error has asked the axis to go to an invalid position. In this case the program error should be corrected.
2. The Command Value field has just been changed and the Motion Controller has not had a chance to acknowledge the new request.

9.1.5.2. Command Velocity

Type:	Axis Status Register
Address:	F*n*:57, where *n* = 8 + the axis number
Format:	Floating Point
Units:	pu/sec

Description

On a Position axis, the Command Velocity has no meaning and is set to the same value as the Target Position.

On a Velocity axis, the Command Velocity is the requested speed.

9.1.5.3. Target Acceleration

Type:	Axis Status Register
Address:	F*n*:55, where *n* = 8 + the axis number
Format:	Floating Point
Units:	pu/sec^2

Description

When the axis is in Closed Loop control, the Target Acceleration is the calculated instantaneous ideal velocity for the axis. The Target Acceleration is calculated every loop by the target generator of the RMC70.

9.1.5.4. Target Position

Type:	Axis Status Register
Address:	F*n*:53, where *n* = 8 + the axis number
Format:	Floating Point
Units:	pu

Description

When the axis is in Closed Loop control, the Target Position is the calculated instantaneous ideal position for the axis. The Target Position is calculated every loop by the target generator of the RMC70. During a move the path of the Target Position toward the Command Position will be the perfect profile for the Actual Position to follow.

The PFID routine uses the difference between the Target Position and the Actual Position to compute any required corrective Control Output.

NOTE:
When an axis is stopped, the Target Position should be the same as the Command Position unless an error or HALT has occurred (see Status).

NOTE:
When an axis is in Open Loop control, the Target Position is set to the Actual Position.

Why Bother?

Knowing the relationship between the Target Position and Actual Position is key to tuning the axis. The main goal in tuning the axis is to minimize the error between the Target and Actual Positions. The plot function is a very useful visual aid in tuning the axis.

9.1.5.5. Target Velocity

Type:	Axis Status Register
Address:	F*n*:54, where *n* = 8 + the axis number
Format:	Floating Point or Special - see below

Units:	pu/sec

Description

When the axis is in Closed Loop control, the Target Velocity is the calculated instantaneous ideal velocity for the axis. The Target Velocity is calculated every loop by the target generator of the RMC70.

9.1.6. Secondary Control

9.1.6.1. Pressure/Force Differential Term

Type:	Axis Status Register
Address:	F*n*:48, where *n* = 8 + the axis number
Format:	Floating Point
Units:	% of maximum control output

Description

The Pressure or Force Differential Term is the portion of the PFID output contributed by the Pressure or Force Differential Gain.

See Pressure/Force Differential Gain for more details.

9.1.6.2. Pressure/Force Error

Type:	Axis Status Register
Address:	F*n*:44, where *n* = 8 + the axis number
Format:	Floating Point
Units:	Pr or F

Description

The Pressure or Force Error is the difference between the Target Pressure or Force and the Actual Pressure or Force. If this value exceeds the Position Error Tolerance while in closed loop control on a position axis, the Pressure/Force Following Error bit will be set.

9.1.6.3. Pressure/Force Feed Forward Term

Type:	Axis Status Register
Address:	F*n*:49, where *n* = 8 + the axis number
Format:	Floating Point
Units:	% of maximum control output

Description

The Pressure/Force Feed Forward Term is the portion of the control output contributed by the Pressure/Force Feed Forward Term.

See Pressure/Force Feed Forward Gain for more details.

9.1.6.4. Pressure/Force Proportional Term

Type:	Axis Status Register
Address:	F*n*:46, where *n* = 8 + the axis number
Format:	Floating Point
Units:	% of maximum control output

Description

The Pressure/Force Proportional Output Term is the portion of control output contributed by the Pressure/Force Proportional Gain.

See Pressure/Force Proportional Gain for more details.

9.1.6.5. Pressure/Force Rate Feed Forward Term

Type:	Axis Status Register
Address:	F*n*:50, where *n* = 8 + the axis number
Format:	Floating Point
Units:	% of maximum control output

Description

The Pressure/Force Rate Feed Forward Term is the portion of control output contributed by the Pressure/Force Rate Feed Forward Gain.

See the Pressure/Force Rate Feed Forward topic for more details.

9.1.7. Secondary Target

9.1.7.1. Command Pressure/Force

Type:	Axis Status Register
Address:	F*n*:61, where *n* = 8 + the axis number
Format:	Floating Point
Units:	Pr or F

Description

The Command Pressure or Force is the requested pressure or force.

9.1.7.2. Target Pressure/Force

Type:	Axis Status Register
Address:	F*n*:60, where *n* = 8 + the axis number
Format:	Floating Point
Units:	Pr or F

Description

When the axis is in closed loop pressure or force control, the Target Pressure/Force is the calculated instantaneous ideal pressure or Force for the axis. The Target Pressure/Force is calculated every loop by the target generator of the RMC70. During a pressure or force

move, the path of the Target Pressure/Force toward the Command Pressure/Force will be the perfect profile for the Actual Pressure/Force to follow.

9.2. Parameter Registers

9.2.1. Axis Parameter Registers Overview

The Axis Parameter Registers contain configuration information for each axis. These registers are editable. Each Register is a 32-bit word. The Parameter Registers are axis or transducer dependant. The list below contains Parameter Registers grouped by axis or transducer type. For details on the addressing format for the registers, see the Register Address Format topic.

For a list of the Parameter Registers, see the Register Map topic.

9.2.2. Feedback

9.2.2.1. Input Filters

9.2.2.1.1. Actual Acceleration Filter

Type:	Axis Parameter Register
Address:	Primary Input: F*n*:4, where *n* = 12 + the axis number
Format:	Floating Point
Units:	Hz
Range:	≥ 0
Default Value:	25

Description

This parameter specifies the cut-off frequency of the Actual Acceleration input filter. The filter is applied to the Actual Acceleration before it is used in the control algorithm. The filter is a low-pass fourth order Butterworth filter.

Setting this value to zero (0) disables the filter.

Why Bother?

Filtering the input can reduce noise in the feedback which may improve system control. It also makes the plots look cleaner.

9.2.2.1.2. Actual Position Filter

Type:	Axis Parameter Register
Address:	F*n*:2, where *n* = 12 + the axis number
Format:	Floating Point
Units:	Hz
Range:	≥ 0
Default Value:	0

Description

This parameter specifies the cut-off frequency of the Actual Position input filter. The filter is applied to the Actual Position before it is used in the control algorithm. The filter is a low-pass fourth order Butterworth filter.

Setting this value to zero (0) disables the filter.

Why Bother?

Filtering the input can reduce noise in the feedback and improve control on a system. It also makes the plots look cleaner.

9.2.2.1.3. Actual Pressure / Force Filter

Type:	Axis Parameter Register
Address:	**Primary Input:** F*n*:4, where *n* = 12 + the axis number
	Secondary Input: F*n*:22, where *n* = 12 + the axis number
Format:	Floating Point
Units:	Hz
Range:	≥ 0
Default Value:	

Description

This parameter specifies the cut-off frequency of the Actual Pressure/Force input filter. The filter is applied to the Actual Pressure/Force before it is used in the control algorithm. The filter is a low-pass fourth order Butterworth filter.

Setting this value to zero (0) disables the filter.

Why Bother?

Filtering the input can reduce noise in the feedback which may improve system control. It also makes the plots look cleaner.

9.2.2.1.4. Actual Pressure / Force Rate Filter

Type:	Axis Parameter Register
Address:	**Primary Input:** F*n*:5, where *n* = 12 + the axis number
	Secondary Input: F*n*:23, where *n* = 12 + the axis number
Format:	Floating Point
Units:	Hz
Range:	≥ 0
Default Value:	

Description

This parameter specifies the cut-off frequency of the Actual Pressure/Force Rate input filter. The filter is applied to the Actual Pressure/Force Rate before it is used in the control algorithm. The filter is a low-pass fourth order Butterworth filter.

Setting this value to zero (0) disables the filter.

Why Bother?

Filtering the input can reduce noise in the feedback which may improve system control. It also makes the plots look cleaner.

9.2.2.1.5. Actual Velocity Filter

Type:	Axis Parameter Register
Address:	F*n*:3, where *n* = 12 + the axis number
Format:	Floating Point
Units:	Hz
Range:	≥ 0
Default Value:	100

Description

This parameter specifies the cut-off frequency of the Actual Velocity input filter. The filter is applied to the Actual Velocity before it is used in the control algorithm. The filter is a low-pass fourth order Butterworth filter.

Setting this value to zero (0) disables the filter.

Why Bother?

Filtering the input can reduce noise in the feedback and improve control on a system.

9.2.2.2. Pressure & Force

9.2.2.2.1. Force A Offset, Force B Offset

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point or Special - see below
Units:	
Range:	
Default Value:	

Description

These parameters are used on differential force axes to shift the transducer counts before scaling them to user force units.

9.2.2.2.2. Force A Scale, Force B Scale

Type:	Axis Parameter Register
Address:	Force A Scale: F*n*:0 Force B Scale: F*n*:2, where *n* = 12 + the axis number
Format:	Floating Point
Units:	
Range:	> 0
Default Value:	1

Description

The Force Scale A and B parameters are used on dual-input force axes with the Force Offset parameter to convert the Transducer Counts to user-defined Actual Force units.

Scale A is used to calculate Actual Force A in force units from Counts A, and Scale B is used to calculate Actual Force B in force units from Counts B. Actual Force B is then subtracted from Actual Force A to give the Actual Force.

9.2.2.2.3. Pressure/Force Offset

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pr/F
Range:	
Default Value:	

Description

This parameter is used on single-input pressure or force axes to shift the transducer counts before scaling them to user force units.

9.2.2.2.4. Pressure/Force Scale

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	
Range:	
Default Value:	

Description

The Pressure or Force Scale is used on single-input pressure or force axes with the Pressure/Force Scale parameter to convert the Transducer Counts to user-defined Actual Force units.

9.2.2.3. Transducer Specific

9.2.2.3.1. Feedback Type

Type:	Axis Parameter Register
Address:	F*n*:10.8, where *n* = 12 + the axis number
Format:	Bits - see below

Description

This parameter is valid only on MA Axis Modules. The Feedback Type parameter specifies the feedback type of the axis. The following options are available:

- **MDT**
  Magnetostrictive Displacement Transducer. Supports Start/Stop rising edge, Start/Stop falling edge, and Pulse-Width Modulated (PWM).
- **SSI**
  Synchronous Serial Interface.

Address Details

This section is primarily for addressing the Feedback Type parameter when communicating with the RMC7*x* from an external device. This information is not necessary when configuring the Feedback in RMC70Tools.

The Feedback Type is selected with bit 8 in register F*n*:10, where *n* = 12 + the axis number. This bit corresponds to the Feedback Type as shown in the following table:

Bit 8	MDT Type
0	MDT
1	SSI

9.2.2.3.2. MDT Type

Type:	Axis Parameter Register
Address:	F*n*:10.0-2, where *n* = 12 + the axis number
Format:	Bits - see below

Description

This parameter is valid only on axes with MDT feedback. The MDT Type parameter specifies the type of Magnetostrictive Displacement Transducer (MDT). The following options are available:

- **St / St Rising**
  The MDT feedback is Start/Stop, measured on the rising edge.
- **St / St Falling**
  The MDT feedback is Start/Stop, measured on the falling edge.
- **PWM**
  The MDT feedback is Pulse Wdith Modulated.

Address Details

This section is primarily for addressing the MDT Type parameter when communicating with the RMC7*x* from an external device. This information is not necessary when configuring the MDT Type in RMC70Tools.

The MDT Type is selected with bits 0-2 in register F*n*:10, where *n* = 12 + the axis number. These bits correspond to the MDT Type as shown in the following table:

Bit 2	Bit 1	Bit 0	MDT Type
0	0	0	Start/Stop Rising
0	0	1	Start/Stop Falling
0	1	0	PWM

SSI Clock Rate

Type:	Axis Parameter Register
Address:	F*n*:10.10-11, where *n* = 12 + the axis number
Format:	Bits - see below

Description

This parameter is valid only on axes with SSI feedback. You must set this parameter to match your SSI transducer or encoder. The clock rate can be set to the following values:

- 150 kHz
- 250 kHz
- 375 kHz

Choosing the SSI Clock Rate

In general, choose the fastest clock rate possible to obtain the most accurate feedback values. Before selecting the clock rate, verify the following is true:

- Your encoder or transducer supports the clock rate. Refer to your encoder documentation.
- The clock rate is correct for your cable length. The following table gives a rough estimate of the maximum cable length for each clock rate:

Note:
The following table is only a rough estimate and will vary depending on the cable type. Refer to your encoder documentation for accurate information.

Clock Rate	Maximum Cable Length
150 kHz	600 ft
250 kHz	360 ft
375 kHz	240 ft

Address Details

This section is primarily for addressing the SSI Clock Rate parameter when communicating with the RMC7*x* from an external device. This information is not necessary when configuring the SSI Clock Rate in RMC70Tools.

The SSI Clock Rate is selected with bits 10-11 in register F*n*:10, where *n* = 12 + the axis number. These bits correspond to the SSI Clock Rate as shown in the following table:

Bit 10	Bit 11	SSI Clock Rate
0	0	375 kHz
0	1	250 kHz
1	0	150 kHz

SSI Format

Type:	Axis Parameter Register
Address:	F*n*:10.9, where *n* = 12 + the axis number
Format:	Bit - see below

Description

This parameter is valid only on axes with SSI feedback. You must set this parameter to match your SSI transducer or encoder. The following options are available:

- **Binary**
- **Gray**

Address Details

This section is primarily for addressing the SSI Format parameter when communicating with the RMC7*x* from an external device. This information is not necessary when configuring the SSI Format in RMC70Tools.

The SSI Format is selected with bit 9 in register F*n*:10, where *n* = 12 + the axis number. This bit corresponds to the SSI Format as shown in the following table:

Bit 9	SSI Format
0	Binary
1	Gray Code

SSI Data Bits

Type:	Axis Parameter Register
Address:	F*n*:10.12-17, where *n* = 12 + the axis number
Format:	Bits - see below

Description

This parameter is valid only on axes with SSI feedback. It tells the RMC70 how many data bits your SSI encoder has. You must set this parameter to match your SSI transducer or encoder. The SSI Data Bits must be an integer number.

Address Details

This section is primarily for addressing the SSI Data Bits parameter when communicating with the RMC7*x* from an external device. This information is not necessary when configuring the SSI Data Bits in RMC70Tools.

The SSI Data Bits is selected with bits 12-17 in register F*n*:10, where *n* = 12 + the axis number. These bits are the binary representation of the number of data bits.

9.2.2.4. Absolute/Incremental

Type:	Axis Parameter Register Bit Parameter
Address:	F*n*:9.1, where *n* = 12 + the axis number
Format:	bit
Default Value:	0 (Absolute)

Description

The Absolute/Incremental Bit Parameter specifies whether the axis is set as an Incremental or Absolute axis or a Linear axis. When this bit is set, the axis is incremental. When this bit is cleared, the axis is absolute.

9.2.2.5. Noise Error Rate

Type:	Axis Parameter Register
Address:	**Primary Input:** F*n*:6, where *n* = 12 + the axis number
	Secondary Input: F*n*:24, where *n* = 12 + the axis number
Format:	Floating Point
Units:	See Below
Range:	≥ 0
Default Value:	100000

Description

This parameter defines the maximum allowable rate of change in the units of the axis. If the rate of change exceeds the **Noise Error Rate** for a period of 10 ms, then the Noise Error bit will be set. If the Noise Error bit becomes set, it may cause a Halt if so configured by the Auto Stops and the axis is not in the Direct Drive state. During noise less than 10 msec, the Actual Position is estimated to allow recovery from electrical noise.

Units

The Noise Error Rate applies to all axes types. However, the Noise Error Rate units for each axis type may differ. The following table shows the Noise Error Rate units for each axis type:

Axis Type	Noise Error Rate Units
Position	pu/s
Velocity	pu/s^2
Pressure/Force	pu/s

9.2.2.6. Position Offset

Type:	Axis Parameter Register
Address:	F*n*:1, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu
Range:	all
Default Value:	0

Description

This parameter is used on position axes together with the Position Scale parameter to convert the transducer feedback Counts to an Actual Position. After the Counts are scaled to position units, this parameter is used to shift the position units.

Setting the Offset

To set the Position Offset, first set the Position Scale. Then, determine the position, P_0, at which the Actual Position should be zero. Set the Position Offset to -P_0. For details on scaling counts to Position Units, see the appropriate topic:

- MA MDT Scaling (MA module)
- MA SSI Scaling (MA module)
- Analog Scaling (AA or AP module)

9.2.2.7. Position Scale

Type:	Axis Parameter Register
Address:	F*n*:0, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/Counts
Range:	any
Default Value:	1

Description

This parameter is used together with the Position Offset parameter to convert the transducer feedback Counts to an Actual Position. The Position Scale specifies how many position units equal one transducer feedback Count.

To reverse the direction of the feedback, use a negative Position Scale.

Setting the Scale

In general, to set the scale, determine how many Counts correspond to one position unit (inches, meters, etc., as desired). The Position Scale is the inverse of that number. For specific details on converting counts to position units refer to the appropriate topic:

- MDT Scaling (MA module)
- SSI Scaling (MA module)
- Analog Scaling (AA or AP module)

Why bother?

It is important to specify the measurement units to be used for each axes. Setting the Scale converts the information from the transducer into meaningful measurement units.

9.2.2.8. Linear/Rotary

Type:	Axis Parameter Register Bit Parameter
Address:	F*n*:9.0, where *n* = 12 + the axis number
Format:	bit
Default Value:	0 (Linear)

Description

The Linear/Rotary Bit Parameter specifies whether the axis is set as a Rotary axis or a Linear axis.

When this bit is set, the axis is Rotary. When this bit is cleared, the axis is linear.

9.2.2.9. Stop Threshold

Type:	Axis Parameter Register
Address:	F*n*:5, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/s
Range:	≥ 0
Default Value:	10

Description

The Stop Threshold specifies a velocity threshold for considering an axis stopped. When the absolute value of the velocity falls below this threshold, the axis is considered stopped, and the Stopped status bit will be set. The Stopped bit is not latched and will clear when the velocity exceeds the threshold.

The Stop Threshold does not apply to axis with only pressure or force control because the axis may be physically moving even thought the pressure does not change. The stopped status of axes with position-pressure or velocity-pressure control is determined by the position or velocity axis.

Why Bother?

If an axis was considered stopped when the velocity is zero, it would be impossible for it to be considered stopped, since there is always noise in real-life systems. For example, a system scaled in inches may have velocities up to 0.1 in/sec when it's standing still. Setting the Stop Threshold to 0.12 in/sec will let you know when the axis is stopped.

9.2.2.10. Velocity Deadband

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	V
Range:	
Default Value:	

Description

This parameter is used on velocity axes to compensate for a deadband in the system.

9.2.2.11. Velocity Offset

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	V
Range:	
Default Value:	

Description

This parameter is used on velocity axes to shift the counts before scaling them to user velocity units.

9.2.3. Halting

9.2.3.1. Auto Stop Configuration

Type: Axis Parameter Register

Address: F*n*:106 to F*n*:108, where *n* = 12 + the axis number

Address Details

The Auto Stop Registers are in 32-bit integer format, not floating point numbers like most of the Parameter Registers. There are 3 Auto Stop registers per axis: F*n*:106, F*n*:107, F*n*:108, where *n* = 8 + the axis number.

Each Auto Stop level is configured with 3 bits. These bits correspond to the Auto Stop Levels as shown in the following table:

High Bit	Mid Bit	Low Bit	Level
0	0	0	Status Only
0	0	1	External Halt
0	1	0	Closed Loop Halt
0	1	1	Open Loop Halt
1	0	0	Open Loop Halt and Disable Output

The bit numbers for each Auto Stop can be found in the Register Map or

in the "Reg #" column in the Axes Parameters Pane in the Axis Tools.

Example

The user would like to configure the Auto Stop for the Axis 0 No Transducer Error Bit as an Open Loop Halt.

1. In the Register Map, the address for the No Transducer Auto Stop is given as F*n*:106.18-20.
2. *n* = 8 + 0 = 8.
3. To configure the Auto Stop as Open Loop Halt, bits 18-20 in the register F8:106 are set as shown in the table:

Bit:	20	19	18
	0	1	1

Format: Special - See Address Details below

Description

Auto Stops are Halts that are defined to occur automatically on rising edges of certain error bits. The user has control over which error bits cause which levels of halts. Auto Stops are always active unless the axis has been issued a Direct Drive (9) command, in which case no Auto Stops will be triggered even though error bits may be set.

Each Auto Stop may be configured to one of the following levels:

Level	Action
Status Only	No action is taken.
External Halt	An External Halt is initiated. An External Halt only sets the External Halt status bit. No other action is taken.
Closed Loop Halt	If the axis is currently in Position or Velocity Closed Loop Control, then a Closed Loop Halt is initiated, otherwise an Open Loop Halt is initiated.
Open Loop Halt	An Open Loop Halt is initiated.
Open Loop Halt and Disable Output	An Open Loop Halt and Disable Output is initiated.

See the Halts topic for details on the steps taken by the RMC70 when a Halt occurs.

Configuring the Auto Stops

To configure the Auto Stops:

- Click the **Axis Tools** button ( ) in the toolbar.
- In the **Axis Parameters** pane, on the **Setup** tab, expand the **Halts** section and expand the **Auto Stop Configuration** section.
- Choose a halt type for each error bit.
- Click the **download** button ( ) in the **Axis Tools** toolbar.

Why Bother?

A good system designer not only considers how a machine reacts in normal operation, but also considers how it reacts when something fails. The Auto Stops help the designer avoid personal injury or equipment damage when something does fail.

Example

A system designer decided to set all the Auto Stops to Status Only to avoid the hassle of having the system stop when he was tuning it up. Unfortunately, the feedback transducer on one of the hydraulic cylinders failed. The RMC thought the cylinder was at 0 inches when it really was at 20 inches.

The feedback error caused the RMC to apply full drive to the cylinder. The cylinder slammed to the end of the stroke, smashing sensitive machine parts. This cylinder had also been synchronized with another cylinder to push a machine part back and forth. Of course, the cylinders went out of synchronization, bending the linkages between them.

If the Auto Stop for the No Transducer Error Bit had been set to Open Loop Halt, this accident could have been avoided. However, if it had been set to Closed Loop halt, the accident would still have occurred, because a Closed Loop Halt still requires feedback.

9.2.3.2. Halt Group Number

Type:	Axis Parameter Register
Address:	F*n*:112, where *n* = 12 + the axis number
Format:	Floating Point
Units:	none
Range:	≥ 0
Default Value:	0

Description

This parameter defines which halt group the axis is in. If any member of a halt group halts, all axes in that group will halt. Zero indicates that the axis is not a member of a halt group.

9.2.3.3. Closed Loop Halt Deceleration

Type:	Axis Parameter Register
Address:	F*n*:110, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/sec^2
Range:	> 0
Default Value:	100

Description

This register specifies the rate at which an axis is decelerated to zero velocity in closed loop control due to a Closed Loop Halt. A Closed Loop Halt can be initiated by the Closed Loop Halt (1) command or an Auto Stop.

Note:
The deceleration specified by the Closed Loop Halt Deceleration is the *average* deceleration. The instantaneous deceleration may exceed this value.

9.2.3.4. Open Loop Halt Ramp

Type:	Axis Parameter Register

Address:	F*n*:111, where *n* = 12 + the axis number
Format:	Floating Point
Units:	V/s
Range:	≥ 0
Default Value:	100

Description

This register specifies the rate at which the output is ramped to zero volts due to an Open Loop Halt. An Open Loop Halt can be initiated by the Open Loop Halt (2) command or an Auto Stop.

9.2.4. Output

9.2.4.1. Deadband Tolerance

Type:	Axis Parameter Register
Address:	F*n*:41, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu
Range:	≥ 0
Default Value:	0

Description

The Deadband Tolerance modifies the Output Deadband. If the Output Deadband is 0, the Deadband Tolerance will have no effect. See the Output Deadband topic for more details.

The Deadband Tolerance specifies a Position Error window inside of which only a fraction of the Output Deadband is applied. This minimizes chattering, which might otherwise be caused by the Output Deadband. If the Position Error is greater than the Deadband Tolerance, the Output Deadband is applied normally. See the Tuning topic for details on how to properly adjust the Deadband Tolerance.

The Deadband Tolerance works as follows:

- If the absolute value of the Position Error is less than the Deadband Tolerance, only a fraction of the Output Deadband, proportional to the Position Error, is applied.
- If the absolute value of the Position Error is greater than the Deadband Tolerance, the full Output Deadband is applied.

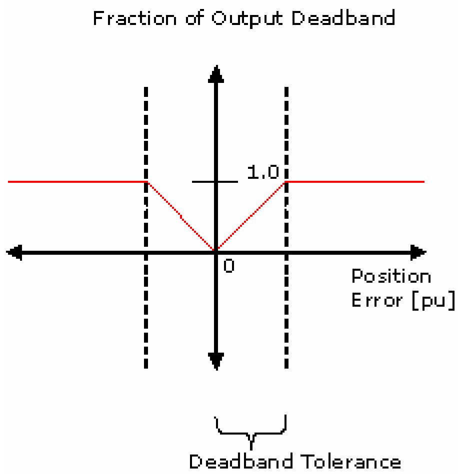


Figure 1: Deadband Tolerance

9.2.4.2. Directional Gain Ratio

Type:	Axis Parameter Register
Address:	F*n*:40, where *n* = 12 + the axis number
Format:	Floating Point
Range:	> 0
Default Value:	1

Description

The Directional Gain Ratio is used to ratio the Control Output depending on the direction of travel. This accounts for system dynamics that change with the direction of travel, as occurs with single-rod hydraulic cylinders. If you are not using a single-rod hydraulic cylinder, you should probably leave this parameter at its default value of 1.

The Directional Gain Ratio will never increase the Control Output. It will only decrease the output in one direction. If the ratio is greater than one, it will decrease the Control Output in the negative direction. If the ratio is less than one, it will decrease the Control Output in the positive direction. The default value of 1 uses the same Control Output in both directions.

The Control Output will be multiplied by the Directional Gain Ratio depending on the direction of travel.

Directional Gain Ratio	Multiplier in Positive Direction	Multiplier in Negative Direction
1	1	1
> 1	1	1 / Directional Gain Ratio
< 1	Directional Gain Ratio	1

Determining the Directional Gain Ratio

In the tuning process, the Directional Gain should be set before adjusting any gains. Follow these steps to set the Directional Gain Ratio:

- Use the Open Loop Rate (10) command to move the axis in the positive direction. Use a value of Requested Output, for example 1.0 V, that will move the axis at a reasonable speed. Make sure the cylinder reaches a constant speed. Be prepared to

stop the axis with a Requested Output of 0 to keep it from slamming into the end of the cylinder. Record the constant Actual Velocity that the cylinder reached.

- Repeat the previous step for the opposite direction (the Requested Output will be negative).
- The Directional Gain Ratio parameter should be: (Velocity in Negative Direction) / (Velocity in Positive Direction)

A single-rod hydraulic cylinder typically requires a Directional Gain Ratio less than 1 (if the positive direction is the extend direction).

9.2.4.3. Invert Output Polarity

Type:	Axis Parameter Register
Address:	F*n*:34.0, where *n* = 12 + the axis number
Format:	Bit
Default:	0 (not inverted)

Description

The Invert Output Polarity parameter bit specifies the polarity of the Control Output.

If the Invert Output Polarity bit is cleared, the physical voltage on the Control Output pins of the RMC70 Axis Module will be of the same polarity as the Control Output. If the Invert Output Polarity bit is set, the Control Output pins of the RMC70 Axis Module will be the negative of the voltage of the Control Output.

Why Bother?

If you wire up the system and find out that a positive Control Output moves the axis in the negative direction, instead of rewiring the Control Output, you can just set this bit. This is often useful on motors, when it isn't always obvious which way it should be wired.

9.2.4.4. Output Bias

Type:	Axis Parameter Register
Address:	F*n*:33, where *n* = 12 + the axis number
Format:	Floating Point
Units:	Volts
Range:	-10 to 10
Default Value:	0

Description

The Output Bias voltage is always added to the Control Output. Use the Output Bias to compensate for hydraulic valves or other systems that have an offset. To set the Output Bias, use the Open Loop Rate (10) command to find out how much Control Output is required to keep the axis from moving. This is called the null drive. Set the Output Bias parameter to that value.

Note:
Usually there is a null or offset adjustment on the valve or amplifier that can be adjusted so the null drive is 0.

9.2.4.5. Output Deadband

Type:	Axis Parameter Register
Address:	F*n*:42, where *n* = 12 + the axis number
Format:	Floating Point
Units:	V
Range:	0 to 10
Default Value:	0 (disabled)

Description

The Output Deadband is the voltage added or subtracted from the Control Output to compensate for a "deadband" in the system. Some valves or drives do not react to small changes in output; this effect is termed "dead band".

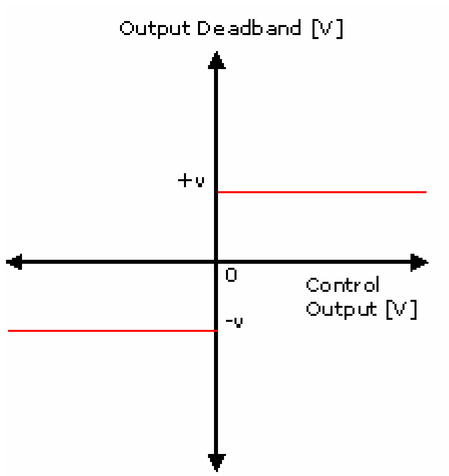
The Output Deadband, works together with the Deadband Tolerance parameter. These are essential tuning parameters if the system exhibits a deadband. Care must be taken to not make this value too large or the axis may oscillate. Normally the Output Deadband value will be less than 2 volts, and often a few hundred millivolts. Many systems do not require deadband compensation.

How it Works

How the Output Deadband works:

The Output Deadband is added to or subtracted from the Control Output (depending on its sign) so that the Control Output is outside the dead band.

- If the Control Output is positive, the Output Deadband (v) is added to the Control Output.
- If the Control Output is negative, the Output Deadband (v) is subtracted from the Control Output.



The Output Deadband also works together with the following parameters:

- **Deadband Tolerance parameter**
  If the Position Error is less than the Deadband Tolerance window, only a fraction of the Control Output, proportional to the Position Error, is applied. This helps keep the axis from chattering. See the Deadband Tolerance topic for details.
- **Integrator**
  The Output Deadband also affects the functionality of the Integral Term. See the Integral Term and Integral Gain topics for details.

Determining the Output Deadband

Follow these steps to determine what the Output Deadband should be set to:

1. Give increasing amounts of Control Output to the system with the Direct Drive (9) command until the system starts to move. Start with a Requested Output of 0 V. The value of Control Output at which the system starts to move is your deadband.
2. If the deadband is less than a few hundred millivolts (approx. 0.4V), your system probably does not need the deadband eliminator. Set the Output Deadband to 0.
3. If the deadband is greater than a few hundred millivolts (approx. 0.4V), you probably need the deadband eliminator. Set the Output Deadband to the value of your deadband.

Why Bother?

On an axis with overlapped spools or a large amount of static friction, the axis may oscillate slowly around the Command Position. This happens when the axis does not respond to small changes in Control Output. The integrator winds up until the axis moves and then has too much Control Output, overshooting the Command Position. The axis then winds the integrator down until the axis overshoots the other way.

If this happens, look at the Control Output. The Control Output will oscillate slowly as the axis oscillates. Notice the peak-to-peak values of the Control Output oscillation and subtract the minimum value from the maximum value and then divide the result by two. Enter this value in the Output Deadband register as a starting point.

A spool that has 10% overlap will require a value of about 1 Volt.

9.2.4.6. Output Filter

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12+ the axis number
Format:	Floating Point
Units:	pu/sec^2
Range:	≥ 0
Default Value:	0

Description

This parameter specifies the cut-off frequency of the low-pass single pole filter that is applied to the Control Output.

Setting this value to zero (0) disables the filter.

Why Bother?

Use the output filter to keep the valve or drive from "humming" which can heat them up or make them fail prematurely. Cranking up the differential gain too high usually causes humming. If you use the output filter, the differential gain can be cranked up without the valve humming and the valve will still respond to true error.

9.2.4.7. Output Limit

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	V
Range:	0 to 10
Default Value:	0

Description

This parameter determines the maximum Control Output. When the Control Output reaches the Output Limit, it will not increase further and the Output Saturated error bit will be set.

> **Note:**
> The actual maximum voltage of the Control Output is the Output Limit *plus* the Output Bias.

9.2.4.8. Output Scale

Type:	Axis Parameter Register
Address:	F*n*:38, where *n* = 12 + the axis number
Format:	Floating Point
Units:	V/100%
Default Value:	10

Description

The Output Scale scales the Control Output to the actual Control Output voltage. This scale is defined as volts per 100% of Control Output. For most systems, this value should be left at its default of 10. For systems that require a Control Output range of anything other than +/- 10 V (or 4-20mA*), this value should be set to the maximum required Control Output voltage. For example, for a system requiring a Control Output range of +/- 5V, set this parameter to 5.

*To convert the RMC70's Control Output from voltage (+/-10V) to current (4-20mA), use the Delta's VC2100 voltage-to-current converter. Information on the VC2100 is available on Delta's website, www.deltamotion.com.

9.2.4.9. Fault Input Polarity

Type:	Axis Parameter Register
Address:	F*n*:34.3, where *n* = 12 + the axis number
Format:	Bit
Default Value:	Active High

Description

This parameter specifies the polarity of the Fault Input. It can be set to the following:

- **Active High**
  When the voltage applied to the Fault Input is greater than 6 volts, the Fault input is active.
- **Active Low**
  When the voltage applied to the Fault Input is less than 6 volts, the Fault input is active.

For details on the Fault Input, see the Fault Input topic.

> **NOTE:**
> The Fault input turns on when a current flows. The polarity of the voltage is unimportant.

9.2.4.10. Enable Output Behavior

Type:	Axis Parameter Register
Address:	F*n*:34.2, where *n* = 12 + the axis number
Format:	Bit
Default Value:	Active Closed

Description

This parameter determines the behavior of the Enable Output on an Axis Module. It can be set to the following:

- **Active Closed**
  When the Enable Output is set, the Enable output switch is closed.
- **Active Open**
  When the Enable Output is set, the Enable output switch is opened.

The Enable Output can be set with the Set Enable Output (67) command. For Enable Output wiring diagrams, see the wiring topic for the specific axis module.

9.2.5. Primary Control

9.2.5.1. In Position Tolerance

Type:	Axis Parameter Register
Address:	F*n*:56, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu
Range:	≥ 0
Default Value:	10

Description

The In Position Tolerance specifies a tolerance around the Command Position. When the Actual Position gets within this window, the In Position status bit is set. The In Position bit is not latched and will clear if the axis moves back outside the In Position window.

The In Position bit not be set if the axis is not In Closed Loop control.

Example

If an axis Command Position is 10.000 and the In Position parameter is 0.030, the In Position bit will be set when the axis is stopped and its Actual Position is between 9.971 and 10.029. The bit will be cleared whenever the Actual Position is outside the range.

9.2.5.2. Position Error Tolerance

Type:	Axis Parameter Register
Address:	F*n*:57, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu
Range:	≥ 0
Default Value:	100

Description

The Position Error Tolerance specifies how large the Position Error may become before the Following Error status bit is set. If the Following Error bit is set, a Halt will occur if the Auto Stops are configured to do so and the axis is not in the Direct Drive state.

9.2.5.3. At Velocity Tolerance

Type:	Axis Parameter Register
Address:	F*n*:58, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/s
Range:	≥ 0
Default Value:	10

Description

The At Velocity Tolerance specifies a tolerance around the Command Velocity. When the Actual Velocity gets within this window, the At Velocity status bit is set. The At Velocity bit is not latched and will clear if the axis speed moves back outside the At Velocity window.

The At Velocity bit not be set if the axis is not In Closed Loop control.

Example

If an axis Command Velocity is 20.000 in/s and the At Velocity Tolerance parameter is 0.30, the At Velocity bit will be set when the axis is at a constant velocity between 19.71 and 20.29. The bit will be cleared whenever the Actual Velocity is outside the range.

9.2.5.4. Jerk Feed Forward

Type:	Axis Parameter Register
Address:	F*n*:67, where *n* = 12 + the axis number
Format:	Floating Point
Units:	%
Range:	0 - 110
Default Value:	100

Note:
In early versions of RMC70 firmware, this register was called Integrator Limit and had a different function. If your controller has an Integrator Limit, please update the firmware.

Description

The Jerk Feed Forward causes the controller to give extra Control Output proportional to the change in acceleration. Acceleration Feed Forward is only applied when the axis is in Closed Loop control and the axis is accelerating. It helps the axis maintain the Target Position. Jerk Feed Forward is helpful in systems that can be modeled as a second-order system. Many systems do not require any Jerk Feed Forward.

The Jerk Feed Forward Term is the portion of the Control Output resulting from the Acceleration Feed Forward Gain.

Tuning

The Jerk Feed Forward helps the axis track the Target Position while the axis is accelerating. This makes it "easier" for the other gains to control the position of the axis. Note that many systems will not need any Jerk Feed Forward.

The Jerk Feed Forward should not be adjusted until the Velocity Feed Forward and Acceleration Feed Forward are adjusted correctly. See Tuning for more details on how to properly adjust the Jerk Feed Forward.

9.2.5.5. Velocity Error Tolerance

Type:	Axis Parameter Register
Address:	F*n*:59, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/s
Range:	≥ 0
Default Value:	100

Description

The Velocity Error Tolerance specifies how large the Velocity Error may become before the Following Error status bit is set. If the Following Error bit is set, a Halt will occur if the Auto Stops are configured to do so and the axis is not in the Direct Drive state.

9.2.5.6. Integrator Mode

Type:	Axis Parameter Register
Address:	F*n*:60.0-3, where *n* = 12 + the axis number
Format:	Bits - see below
Units:	none
Range:	Always Active, Always Held
Default Value:	Always Active

Description

The Integrator Mode specifies the functionality of the Integrator. The following options are available:

- **Always Active**
  The Integrator is always active.
- **Always Held**
  The Integrator will be held at the value at which it was when the Always Held mode is selected.

Format Details

This section is primarily for addressing the Integrator Mode when communicating with the RMC7*x* from an external device. This information is not necessary when configuring the Integrator Mode in RMC70Tools.

The Integrator Mode is selected with bits 0-3 in register F*n*:60, where *n* = 12 + the axis number. These bits correspond to the Integrator Mode as shown in the following table:

Bit 3	Bit 2	Bit 1	Bit 0	Integrator Mode
0	0	0	0	Always Held
0	0	0	1	Always Active

9.2.5.7. Position Proportional Gain

Type:	Axis Parameter Register
Address:	F*n*:61, where *n* = 12 + the axis number
Format:	Floating Point
Units:	%/pu
Range:	≥ 0
Default Value:	0

Description

The Position Proportional Gain controls how much of the Control Output is generated due to the Position Error on a Position axis.

Several gains contribute to the Control Output on an axis. The contribution from the Proportional Gain is called the Proportional Term. The Position Proportional Gain controls how much Proportional Term is generated proportional to the Position Error on a position axis. The Position Error is defined as the Target Position minus the Actual Position.

Limit

The Proportional *Term* is limited at 100% of the maximum Control Output (10V). The smallest average error that will cause the Proportional Term to be 100% is $e_{avg} = 1/K_p$.

Example:

If the Proportional Gain is set to 0.05, the smallest average error to achieve a Proportional Term of 1 is 1 / 0.05 = 20 pu. The Control Output at this point will be 10 V. Any errors larger than 20 pu will cause the Proportional Term to try to exceed 20 pu, and will set the Output Saturated error bit.

Tuning

The Proportional Gain is an important tuning parameter. It affects the responsiveness of the system. A low gain makes the system sluggish and unresponsive. A gain that is too high makes the axis oscillate or vibrate.

See Tuning for details on how to properly adjust the Proportional Gain.

Mathematical Definition

The Proportional Term is mathematically defined as follows:

$$D_{P_n} = \left(\frac{e_n + e_{n-1}}{2} \right) K_P$$

where

$$D_{P_n} = \text{Proportional Term} \left[\text{\% of maximum Control Output}\right]$$

$$e_n = \text{Position Error at sample n} \left[\text{pu}\right]$$

$$K_P = \text{Proportional Gain} \left[\text{\%/pu}\right]$$

The Proportional Gain units are: percent of the maximum Control Output per position unit (pu) of Position Error. The maximum Control Output is 10V.

As illustrated in the equation, the Proportional Term is proportional to the *average* of the last two position errors. This is because a trapezoidal approximation of the Position Error is used.

9.2.5.8. Position Integral Gain

Type:	Axis Parameter Register
Address:	F*n*:62, where *n* = 12 + the axis number
Format:	Floating Point
Units:	%/pu·sec
Range:	≥ 0
Default Value:	0

Description

The Position Integral Gain causes the Integral Term to increase proportional to the integral (sum of size x length of time) of a position error on the axis. The Integral Gain helps compensate for changing system dynamics, such as varying loads, and often aids the axis in rapidly reaching the Command Position. A low gain slows the response of the axis to changes and may keep the axis from reaching the Command Position. A gain that is too high may make the axis oscillate. See Tuning for details on how to properly adjust the Integral Gain.

Several gains contribute to the Control Output on an axis. The contribution from the Integral Gain is called the Integral Term. Each control loop, the Integral Gain is multiplied by the Position Error and added to the Integral Term.

The Integral Gain may be disabled or only be allowed to decrease the Integral Term under certain conditions. See below for more details.

Mathematical Definition

The Integral Term is mathematically defined as follows:

$$D_{I_n} = \left(\frac{e_n + e_{n-1}}{2}\right) K_I T + D_{I_{n-1}}$$

where

$$D_{I_n} = \text{Integral Term}\left[\text{\% of maximum Control Output}\right]$$

$$e_n = \text{Position Error at sample n}\left[\text{pu}\right]$$

$$K_P = \text{Proportional Gain } \left[\text{\%/(pu-sec)}\right]$$

$$T = \text{Time period of control loop}\left[\text{sec}\right]$$

The Integral Gain units are: percent of the maximum Control Output per position unit (pu) of Position Error times time. The maximum Control Output is 10V.

As illustrated in the equation, the Integral Gain is multiplied with the *average* of the last two position errors. This is because a trapezoidal approximation of the Position Error is used.

Special Conditions

Disable Conditions

The Integral Term will be applied, but will not change during the following conditions:

- **Integrator Mode**
  The Integral Gain can be disabled by the user by setting the Integrator Mode to Always Held. The Integrator Term will be held at the value at the time it was set.
- **Integral Limit**
  The Integral Gain is disabled when the Integral Term reaches the Integral Limit as set by the user. When the Integral Term becomes less than the Integral Limit, it will be enabled again. The maximum allowable Integral Limit is 1.0.

- **Control Output Limit**
  The Integral Gain is disabled if the sum of the Integral Term and the other terms (Proportional, Differential, Feed Forwards) that contribute to the Control Output is greater than 110% of the maximum Control Output before output scaling. This helps avoid unnecessarily saturating the Integrator when there is a large position error and the other terms are large, in which case they should be able to decrease the position error.

  Example:

  If the Proportional Term is 0.6, the Differential Term is 0.2, the Velocity Feed Forward is 0.1, and the Acceleration Feed Forward is 0, then the Integral Term will be limited at 1.1 - 0.6 - 0.2 - 0.1 - 0 = 0.2.

Decreasing Only Conditions

The Integral Term will only be allowed to decrease during the following conditions:

- **Within Deadband Window**
  When the Control Output is less than the Output Deadband, the Integral Term will only be allowed to decrease. In systems with a deadband, this avoids ratcheting the axis back and forth around the deadband. See deadband for more details.
- **Within one-half count of Command Position**
  When the axis is within one-half count of the Command Position, the Integral Gain is only allowed to decrease the Integral Term. This is similar to the deadband case. Except in the rare case when the axis is scaled such that the transducer feedback counts match up exactly with the position units, there will always be a small difference between the Command Position and the nearest feedback count. Normally, this would cause the axis to "hunt" for the command position. Constraining the Integral Term to only decrease will keep the axis from "hunting".

9.2.5.9. Position Differential Gain

Type:	Axis Parameter Register
Address:	F*n*:63, where *n* = 12 + the axis number
Format:	Floating Point
Units:	%/(pu/sec)
Range:	≥ 0
Default Value:	0

Description

The Position Differential Gain controls how much of the Control Output is generated due to the rate of change between the target and actual positions on a Position axis.

Several gains contribute to the Control Output on an axis. The contribution from the Differential Gain is called the Differential Term. The Position Differential Gain controls how much Differential Term is generated proportional to the rate of change between the target and actual positions on a position axis.

Definition

The Differential Term is mathematically defined as follows:

$$D_{Dn} = \left(V_{Target} - V_{Actual}\right) K_D$$

where

$$D_{D_n} = \text{Differential Term}\left[\text{\% of maximum Control Output}\right]$$

$$V = \text{Velocity}\left[\text{pu/s}\right]$$

$$K_D = \text{Differential Gain}\left[\text{\%/(pu/s)}\right]$$

The Differential Gain units are: percent of the maximum Control Output per position units per second (pu/s). This is abbreviated as OF/(pu/s). The maximum Control Output is 10V.

Tuning

The Differential Gain may greatly enhance performance on many motion systems. On velocity drives or hydraulic systems, Differential Gain will tend to dampen out oscillations and help the axis track during acceleration and deceleration. On torque drives, the differential gain is essential for providing damping to the motor.

A disadvantage of Differential Gain is that it amplifies position measurement noise. If there is too much noise or the gain is too high, this can cause the system to chatter or oscillate.

See Tuning for details on how to properly adjust the Differential Gain.

9.2.5.10. Velocity Feed Forward

Type:	Axis Parameter Register
Address:	F*n*:66, where *n* = 12 + the axis number
Format:	Floating Point
Units:	%/(pu/s)
Range:	≥ 0
Default Value:	0

Description

Velocity Feed Forward causes the controller to give extra Control Output while moving at constant velocity. Velocity Feed Forward is only applied when the axis is in Closed Loop control. It helps the axis maintain the Target Position while the axis is moving at constant velocity.

The Velocity Feed Forward *Term* is the portion of the Control Output resulting from the Velocity Feed Forward, which is a gain.

Definition

The Velocity Feed Forward is defined as follows:

$$D_{VFF_n} = K_{VFF} V_{Target}$$

where

$$D_{VFF_n} = \text{Velocity Feed Forward Term}\left[\text{\% of maximum Control Output}\right]$$

$$K_{VFF} = \text{Velocity Feed Forward }\left[\text{\%/(pu/s)}\right]$$

$$V_{Target_n} = \text{Target Velocity at sample n}\left[\text{pu/s}\right]$$

Tuning

The Velocity Feed Forward helps the axis maintain the Target Position while the axis is moving. This makes it "easier" for the other gains to control the position of the axis. See Tuning for details on how to properly adjust the Velocity Feed Forward.

9.2.5.11. Acceleration Feed Forward

Type:	Axis Parameter Register
Address:	F*n*:66, where *n* = 12 + the axis number
Format:	Floating Point
Units:	%/(pu/s^2)
Range:	≥ 0
Default Value:	0

Description

The Acceleration Feed Forward causes the controller to give extra Control Output while accelerating or decelerating. Acceleration Feed Forward is only applied when the axis is in Closed Loop control and accelerating or decelerating. It helps the axis maintain the Target Position.

The Acceleration Feed Forward Term is the portion of the Control Output resulting from the Acceleration Feed Forward Gain.

Definition

The Velocity Feed Forward is defined as follows:

$$D_{AFFn} = K_{AFF} A_{Target}$$

where

$$D_{AFF_n} = \text{Acceleration Feed Forward Term}\left[\text{\% of maximum Control Output}\right]$$

$$K_{AFF} = \text{Acceleration Feed Forward }\left[\text{\%/(pu/s)}\right]$$

$$A_{Target_n} = \text{Target Acceleration at sample n}\left[\text{pu/s}\right]$$

Tuning

The Acceleration Feed Forward helps the axis track the Target Position while the axis is accelerating. This makes it "easier" for the other gains to control the position of the axis.

The Acceleration Feed Forward should not be adjusted until the Velocity Feed Forward is adjusted correctly. See Tuning for more details on how to properly adjust the Acceleration Feed Forward.

9.2.6. Simulation

9.2.6.1. Simulate Mode

Type:	Axis Parameter Register
Address:	F*n*:116.0, where *n* = 12 + the axis number
Format:	bit

Description

The RMC70 can simulate a physical system on any axis. During simulate mode, the axis's physical output is 0 V. Use Simulate mode to practicing tuning and moving an axis before doing it on a real system. You can use Simulate mode for simple purposes, such learning how to move an axis, or you can calculate the Simulate parameters such that the simulator approximates a real hydraulic system. Tuning your simulated system familiarizes you with tuning a your hydraulic system.

Simulating a Generic System

To simulate an axis for practicing using RMC70:

- Set the Simulate parameters for the axis to the following:

Parameter	Value:
Simulate Mode	On (checked)
System Gain	10
Natural Frequency	20
Damping Factor	0.8

- Set the Negative Travel Limit to 0, and the Positive Travel Limit to 50.
- Set the tuning parameters to the following:

Parameter	Value:
Proportional Gain	80
Integral Gain	100
Differential Gain	1
Velocity Feed Forward	0.9
Acceleration Feed Forward	0.01

- Issue the Initialize Axis (7) command to the axis.
- You can now issue move commands to the axis.

Simulating a Real System

Use the guidelines below to calculate the Simulate parameters to approximate your real system:

Parameter	To calculate this parameter:
System Gain	The System Gain units are pu/s/V, which is the speed the system moves for 1 V of control output. To find this value, give 1 volt of Control Output (use the Open Loop Rate (10) command). Use the plot to find the steady-state Actual Velocity of the system. This is your System Gain.
Natural Frequency	The Natural Frequency for a hydraulic system is normally between 1 and 30. Use the following formula to calculate the natural frequency for hydraulic system: ω = sqrt[(4*200000 *A^2) /(mass * volume)]

where

ω = Natural Frequency (Hz)

A = area of the piston (in^2)

mass = the mass moved by the system (lb)

volume = the volume of trapped oil in the cylinder (in^3)

Damping Factor: The damping factor is a unitless number. Hydraulic systems typically range from 0.3 to 0.8. If the load has a lot of friction, this value will become larger. A lower value makes the system more difficult to control.

Once you have calculated the tuning parameters, you must tune the axis according to the Tuning Overview in order to move it.

9.2.6.2. System Gain

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/sec^2
Range:	> 0
Default Value:	1

Description

This parameter is used for the motion simulator in the RMC70. For details on using simulate mode, see the Simulate Mode topic.

The System Gain is pu/s/V, in other words, how fast the system moves for 1 V of control output. To find this value, give 1 volt of Control Output (use the Open Loop Rate (10) command). Use the plot to find the Actual Velocity of the system. This is your System Gain.

9.2.6.3. Natural Frequency

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/sec^2
Range:	≥ 0
Default Value:	20

Description

This parameter is used for the motion simulator in the RMC70. For details on using simulate mode, see the Simulate Mode topic.

The Natural Frequency for a hydraulic system is normally between 1 and 30. Use the following formula to calculate the natural frequency for hydraulic system:

$$\omega = \text{sqrt}[(4*200000 *A^2) /(\text{mass} * \text{volume})]$$

where

A = area of the piston (in^2)

mass = the mass moved by the system (lb)

volume = the volume of trapped oil in the cylinder (in^3)

9.2.6.4. Damping Factor

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/sec^2
Range:	≥ 0
Default Value:	0.75

Description

This parameter is used for the motion simulator in the RMC70. For details on using simulate mode, see the Simulate Mode topic.

The damping factor is a unitless number that specifies how much friction there is in the system. Hydraulic systems typically range from 0.3 to 0.8. If the load has a lot of friction, this value will become larger. A lower value makes the system more difficult to control.

9.2.7. Secondary Control

9.2.7.1. At Pressure/Force Tolerance

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/sec^2
Range:	≥ 0
Default Value:	

Description

NOTE
This parameter is valid for Pressure or Force axes. It is called At Pressure Tolerance for pressure axes and At Force Tolerance for force axes.

This parameter specifies a tolerance around the Command Pressure or Command Force. When the Actual Pressure or Force is within this tolerance, the At Pressure/Force Status bit is set. The At Pressure/Force bit is not latched and will be cleared if the axis moves outside the At Pressure/Force Tolerance.

9.2.7.2. Pressure/Force Error Tolerance

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/sec^2
Range:	≥ 0
Default Value:	

Description

NOTE:
This parameter is valid for Pressure or Force axes. It is called Pressure Error Tolerance for pressure axes and Force Error Tolerance for force axes.

This parameter specifies how large the Pressure/Force Following Error may become before the Pressure Error bit is set. The Axis must be in closed loop pressure control for the Pressure Following Error bit to be set. If the Pressure Following Error bit is set, a Halt will occur if the Auto Stops are configured to do so and the axis is not in the Direct Drive state.

9.2.7.3. Pressure/Force Proportional Gain

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/sec^2
Range:	≥ 0
Default Value:	0

Description

NOTE
This parameter is valid for Pressure or Force axes. It is called Pressure Proportional Gain for pressure axes and Force Proportional Gain for force axes.

The Pressure/Force Proportional Gain controls how much of the Control Output is generated due to the Pressure or Force Error on a Pressure or Force axis.

Several gains contribute to the Control Output on an axis. The contribution from the Proportional Gain is called the Proportional Term. The Position Proportional Gain controls how much Proportional Term is generated proportional to the Position Error on a position axis. The Position Error is defined as the Target Position minus the Actual Position.

9.2.7.4. Pressure/Force Integral Gain

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/sec^2
Range:	≥ 0

Default Value:	0

Description

NOTE
This parameter is valid for Pressure or Force axes. It is called Pressure Integral Gain for pressure axes and Force Integral Gain for force axes.

The Pressure/Force Integral Gain causes the Integral Term to increase proportional to the integral (sum of size x length of time) of a position error on the axis. The Integral Gain helps compensate for changing system dynamics, such as varying loads, and often aids the axis in rapidly reaching the Command Pressure/Force. A low gain slows the response of the axis to changes and may keep the axis from reaching the Command Position. A gain that is too high may make the axis oscillate. See Tuning for details on how to properly adjust the Integral Gain.

Several gains contribute to the Control Output on an axis. The contribution from the Integral Gain is called the Integral Term. Each control loop, the Integral Gain is multiplied by the Position Error and added to the Integral Term.

9.2.7.5. Pressure/Force Differential Gain

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/sec^2
Range:	≥ 0
Default Value:	0

Description

NOTE
This parameter is valid for Pressure or Force axes. It is called Pressure Differential Gain for pressure axes and Force Differential Gain for force axes.

The Differential Gain controls how much of the Control Output is generated due to the rate of change of the Pressure/Force Error.

Several gains contribute to the Control Output on an axis. The contribution from the Differential Gain is called the Differential Term. The Position Differential Gain controls how much Differential Term is generated proportional to the rate of change between the target and actual positions on a position axis.

The Pressure/Force Differential *term* is the portion of the Control Output resulting from the Pressure/Force Differential gain.

9.2.7.6. Pressure/Force Feed Forward

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/sec^2
Range:	≥ 0

Default Value:	0

Description

> **NOTE**
> This parameter is valid for Pressure or Force axes. It is called Pressure Feed Forward for pressure axes and Force Feed Forward for force axes.

The Pressure/Force Feed Forward is similar to the Velocity Feed Forward. The Pressure Feed Forward causes the controller to give extra Control Output while the pressure or force is changing at a constant rate. Pressure Feed Forward is only applied when the axis is in Closed Loop control. It helps the axis maintain the Target Pressure/Force while the pressure is changing at a constant rate.

The Pressure/Force Feed Forward Term is the portion of the Control Output resulting from the Velocity Feed Forward, which is a gain.

9.2.7.7. Pressure/Force Rate Feed Forward

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu/sec^2
Range:	≥ 0
Default Value:	0

Description

> **NOTE**
> This parameter is valid for Pressure or Force axes. It is called Pressure Rate Feed Forward for pressure axes and Force Rate Feed Forward for force axes.

The Pressure/Force Rate Feed Forward is similar to the Acceleration Feed Forward. The Pressure/Force Rate Feed Forward causes the controller to give extra Control Output when the rate of change in pressure is changing. Acceleration Feed Forward is only applied when the axis is in Closed Loop control and accelerating or decelerating. It helps the axis maintain the Target Position when the rate of change in pressure is changing

The Pressure/Force Rate Feed Forward Term is the portion of the Control Output resulting from the Acceleration Feed Forward Gain.

9.2.7.8. Pressure/Force Orientation

Type:	Axis Parameter Register
Address:	F*n*:80.5-6, where *n* = 12 + the axis number

Format:	Bit
Values:	Push, Pull, Bi-directional
Default:	Push

Description

> **NOTE**
> This parameter is valid for Pressure or Force axes. It is called Pressure Orientation for pressure axes and Force Orientation for force axes.

This register defines the orientation of the pressure relative to the orientation of the position or velocity. It is important that this parameter is set correctly so that the Control Output will be the correct polarity.

Orientation	Description
Push	The pressure increases as the position units increase.
Pull	The pressure decreases as the position units increase.
Bi-directional	The pressure increases in both directions (starting from 0).

9.2.8. Target

9.2.8.1. Positive Travel Limit

Type:	Axis Parameter Register
Address:	F*n*:92, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu
Range:	any
Default Value:	0

Description

The Positive Travel Limit and the Negative Travel Limit specify the position boundaries in which the axis is allowed to operate. These limits are only valid on Linear axes. The Positive Travel Limit must be greater than or equal to the Negative Travel Limit.

If the Actual Position exceeds the Positive Travel Limit, the Positive Overtravel error bit will be set. If the Actual Position is less than the Negative Travel Limit, the Negative Overtravel error bit will be set. The Overtravel error bits will remain set until the axis moves back inside the limits. The error bit will cause a Halt to occur if the Auto Stops are configured to do so and the axis is not in the Direct Drive state.

> **Note:**
> If the axis is in the Direct Drive (9) state, the Overtravel bits will not be set.

Any motion command issued with a Requested Position that is outside of the travel limits will be set to the closest travel limit, and the Command Modified error bit will be set. The error bit will cause a Halt to occur if the Auto Stops are configured to do so and the axis is not in the Direct Drive state. If the axis is outside of the travel limits when a move command is issued, the axis will only be allowed to move in the direction that brings it closer to the limit.

9.2.8.2. Negative Travel Limit

Type:	Axis Parameter Register
Address:	F*n*:93, where *n* = 12 + the axis number
Format:	Floating Point
Units:	pu
Range:	any
Default Value:	0

Description

The Negative Travel Limit and the Positive Travel Limit specify the position boundaries in which the axis is allowed to operate. These limits are only valid on Linear axes. The Negative Travel Limit must be greater than or equal to the Negative Travel Limit.

If the Actual Position is less than the Negative Travel Limit, the Negative Overtravel error bit will be set. If the Actual Position exceeds the Positive Travel Limit, the Positive Overtravel error bit will be set. The Overtravel error bits will remain set until the axis moves back inside the limits.

Note:
If the axis is in the Direct Drive (9) state, the Overtravel bits will not be set.

Any motion command issued with a Requested Position that is outside of the travel limits will be set to the closest travel limit, and the Command Modified error bit will be set. If the axis is outside of the travel limits when a move command is issued, the axis will only be allowed to move in the direction that brings it closer to the limit.

The error bits will cause a Halt to occur if the Auto Stops are configured to do so and the axis is not in the Direct Drive state.

9.2.8.3. Positive Pressure/Force Limit

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number
Format:	Floating Point
Units:	Pr/F
Range:	any
Default Value:	0

Description

The Positive Pressure/Force Limit and the Negative Pressure/Force Limit specify the pressure or force boundaries in which the axis is allowed to operate. The Positive Pressure/Force Limit must be greater than or equal to the Negative Pressure/Force Limit.

9.2.8.4. Negative Pressure/Force Limit

Type:	Axis Parameter Register
Address:	F*n*:xxx, where *n* = 12 + the axis number

Format:	Floating Point
Units:	Pr/F
Range:	any
Default Value:	0

Description

The Negative Pressure/Force Limit and the Positive Pressure/Force Limit specify the pressure or force boundaries in which the axis is allowed operate. The Negative Pressure/Force Limit must be less than or equal to the Positive Pressure/Force Limit.

10.Communication Types

10.1. RMC70 Communications Overview

Numerous communication protocols are available on the RMC70. This allows almost any external controller, such as a PLC, HMI, personal computer, etc. to control the RM70 and easily integrate it into the rest of the application.

Communications Types

The RMC70 series supports the communication types listed below. Each RMC70 CPU Module has the communication built in. For details on how to set up and use each communication protocol, click on the respective link.

Communication Protocol	Required RMC70 CPU Module
Serial • DF1 (Full- and Half-Duplex) • Modbus/RTU	RMC75S
PROFIBUS-DP	RMC75P

RS-232 Monitor Port

In addition to its basic communication type, each RMC75 has an RS-232 Monitor port. This port is primarily for connecting to a PC running RMC70Tools, but it can also be used to communicate via serial DF1 to other devices.

Register Map

The register map is the core of the RMC70 communications. It gives the addresses for all the information available in the RMC70. When communicating to the RMC70 from a PLC or other host controller, the register map is a valuable reference.

Indirect Data Map

The Indirect Data Map is important for several communication types. It maps data from anywhere in the RMC70 to an array. This structure allows packing of otherwise discontiguous data together for efficient I/O and messaging communications.

10.2. Indirect Data Map

The Indirect Data Map maps data from anywhere in the RMC70 to an array. The intent of this structure is to allow packing of otherwise discontiguous data together for efficient I/O and messaging communications.

Structure

The Indirect Data Map consists of two arrays of values. The first array is called the Indirect Data Map, and holds register addresses that each entry in the second array (Indirect Data) represents. Each of these arrays is 32 registers. The Indirect Data Map can be used for both readable and writable registers.

See the Indirect Data Map Editor topic for details on how to edit it in RMC70Tools.

Indirect Data Map Registers

Indirect Data Map: F17:0 to F17:31

Indirect Data: F18:0 to F18:31

> Note:
> Registers F18:0-31 contain the Indirect Data. These are the registers that should be written or read when communicating to the RMC70, *not* registers F17:0-31.

Indirect Data Map	Indirect Data
F17:0	F18:0
F17:1	F18:1
•	•
•	•
•	•
F17:31	F18:31

10.3. RS-232 Monitor Port

Every RMC70 CPU module has a DTE DB9 serial port labeled "RS-232 Monitor". RMC70Tools uses it to communicate to the RMC70 with the DF1 serial protocol. This serial port may also be used to communicate with other devices via DF1.

Connecting from RMC70Tools to the RMC70

Use these steps to connect from RMC70Tools to the RMC70 using the monitor port:

1. Connect a null-modem cable from the PC to the RS-232 Monitor port on the RMC70. See the **Cable** section in this topic for cable details.
2. Open RMC70Tools.
3. In the Project pane, right-click the controller you wish to connect to and click **Properties**.
4. Click the **Connection** tab.
5. Select the serial port you are using on the PC.
6. Click OK.
7. In the Project pane, right-click the controller and click **Go Online**.

Monitor Port Settings

The monitor port uses the serial DF1 full-duplex protocol.

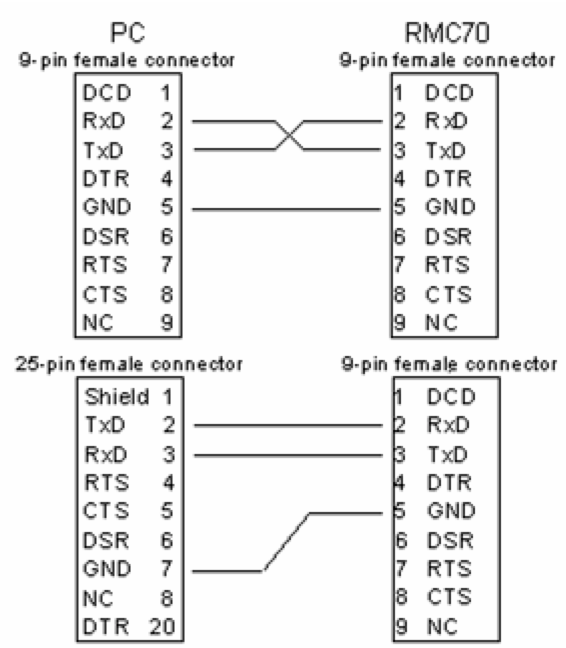
The communication settings of the monitor port are:

- 38400 Baud
- 8 Data Bits
- No Parity
- 1 Stop Bit
- CRC

Cable

The following options are available for the cable between the PC and the RMC serial port:

- Generic null-modem (female on both ends) serial cable
- Allen-Bradley PLC serial cable
- Modicon Modbus cable with a 9-pin male-to-female gender changer
- Siemens SIMATIC 505 Programmable Controller serial cable
- Any cable will work if pins 2, 3, and 5 on the RMC-end of the cable are connected as shown below.



Note:
The DB9 shell is connected to the RMC70 Case.

Communicating with HMIs

The RMC70 can communicate with many types of HMIs (Human-Machine Interfaces). The RMC70 can communicate with an HMI at the same time as it communicates with another system, such as a PLC or RMC70Tools.

HMI Requirements

All RMC75 series motion controllers have an RS-232 Monitor port with the DF1 full-duplex serial protocol. Therefore, if the HMI supports the RS-232 serial DF1 protocol, you can connect the HMI to the RS-232 Monitor port on the RMC70, regardless of the communication type of the RMC70. However, the RMC70 can communicate with HMIs via any of the communication types listed in the Communications Overview topic.
If you want to use RMC70Tools at the same time as an HMI with a serial protocol, you must have two serial ports on the RMC70, which is only available on the RMC705S.

Using the DF1 will simplify the addressing in the HMI, since the register addresses in RMC70Tools are displayed in the DF1 format.

Use Delta's Sample Programs

Delta's web site provides sample HMI programs for communicating with the RMC70. These programs demonstrate reading and writing registers and issuing commands. Using the demo programs can drastically reduce your development time.

Configuring the HMI Communications

For HMIs with serial DF1 support:

Set up the HMI communications as if it were talking to a PLC. When selecting the protocol, choose an Allen-Bradley SLC 500 PLC with DF1.

If you are connecting the HMI to the RS-232 Monitor port on the RMC70, make sure the PLC communication settings are identical to the RS-232 Monitor Port settings.

If you are connecting the HMI to the communication serial port on the RMC70, make sure you configure the serial settings identically to the PLC communication settings.

For HMIs that do not support serial DF1:

Many HMIs require you to choose a PLC and protocol . Although the RMC70 is not a PLC, select any PLC that has a protocol compatible with the RMC70. Make sure to configure the RMC70 serial settings identically to the PLC communication settings.

Using the Communications

General

Once you have set up the HMI, read and write to registers in the RMC70 just as you would with a PLC. For details, see the Serial Overview topic. Use the Register Map to find the addresses of the RMC70 registers.

Issuing Commands

For details on issuing commands from the HMI via a serial protocol, see the Issuing Commands topic.

10.4. PROFIBUS

10.4.1. PROFIBUS-DP Communications Overview

PROFIBUS-DP is the communication protocol available on the RMC7*x*P. PROFIBUS-DP is an open industrial fieldbus and is vendor independent, allowing a large range of PLCs and other PROFIBUS masters to control the RMC70. The RMC70 performs as a PROFIBUS-DP slave,

requiring a PROFIBUS-DP master to control it. The type of the actual master- whether a PLC or other controller - is unimportant. All will communicate with the RMC70 in the same manner.

RMC70 PROFIBUS-DP Specifications

- Operating baud-rates: 9.6Kbaud to 12Mbaud
- Manufacturer Identifier Number: 0x07E1
- Supported: Freeze mode, sync mode, automatic baud-rate detect.
- Modularity: the RMC70 is a modular station that can have one module selected at a time.

RMC70 PROFIBUS-DP Modes

The following PROFIBUS modes are available on the RMC70:

Mode	Commands per Cycle	Simultaneous Commands	Cyclic Read Registers	Explicit Read Registers	Explicit Write Registers
Mode 8-1	1	✓	8	1	1
Mode 16-1	1	✓	8	1	1

For details on reading, writing and issuing commands, see the mode type you are using. For instructions on how to select a mode, see the PROFIBUS Configuration topic.

PROFIBUS-DP CPU Module Panel

The front panel of the RMC*7x*P CPU module includes the following items specific to PROFIBUS communications.

- **9-Pin Connector**
  This connector is the standard PROFIBUS connector. It is used for connecting the RMC70 to other PROFIBUS-DP devices via a standard PROFIBUS-DP cable. The pin assignment is defined by the PROFIBUS specification.
- **LED**
  The LED is illuminated green when the RMC70 is communicating with the PROFIBUS-DP master. During normal operation, this LED should be solid with no flickering.
- **Station Address Switches**
  The 2 rotary switches are used to set the station address. The PROFIBUS Configuration topic describes how to set the station address.

See also:

PROFIBUS Configuration

10.4.2. PROFIBUS Configuration

The following steps are required to connect an RMC70 unit to a PROFIBUS network:

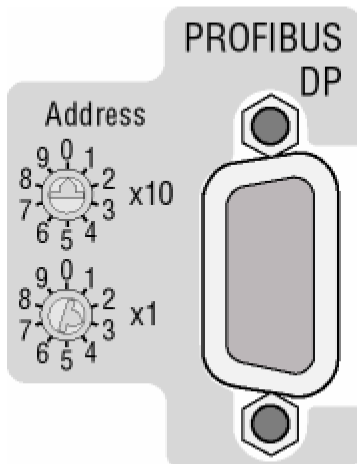
1. Set the RMC's station address

The 2 rotary switches on the front panel of the RMC7*x*P indicate the station address. To set the address, use a small screwdriver to turn the rotary switches to the desired values. The address is the sum of the value of each switch times its respective multiplier.

The RMC7*x*P can use any station address between 1 and 99. Zero (0) is not allowed as a station address. Since the station address 1 is normally used for the master, Delta recommends not setting the address to 1.

The station address on the RMC7*x*P must match the station address as expected by the master. Step 3 explains how to set the station address in the master.

Example



The station address is:
(0 x 10) + (3 x 1) = 3

2. Determine the Appropriate GSD Configuration Module

A GSD file is a device description file used to inform the PROFIBUS master how to communicate with the device. The master requires a GSD file from each device in the network. The RMC7*x*P's GSD file is named **DELT07E1.GSD** and is installed by the RMC70Tools software package. The GSD file is also available from Delta's website www.deltamotion.com.

The DELT07E1.GSD file contains several configuration module entries. When adding an RMC7*x*P to a PROFIBUS network (Step 3), you will need to select exactly one of these configuration modules. Each of these modules is assigned a title, which is often referred to by PROFIBUS configuration software as an *Order Number*.

To determine the correct GSD order number, first determine the operation mode. The following operation modes are available on the RMC7*x*P:

- Basic with 8 Input Registers
- Basic with 16 Input Registers
- Basic with 7-Register Read/Write
- Basic with 31-Register Read/Write

If you do not know the difference between these modes or have not yet determined which is right for your application, please read operation mode topic.

3. Configure the PROFIBUS Network

The PROFIBUS network configuration is stored in the master PROFIBUS device. Creating this configuration is the most difficult step in making the network work. This step requires a device description file (GSD file) from each device in the

network; the GSD file for the RMC70 is named **DELT07E1.GSD** and is installed by the RMC70Tools software package.

The following steps outline configuring your PROFIBUS-DP network. The steps are general because several master configuration programs are available, and each handles these steps differently.

1. Open your PROFIBUS-DP master configuration program.
2. If you are modifying an existing PROFIBUS-DP network, open you current configuration file.
3. If you are creating a new PROFIBUS-DP network, you must create a new network, add a master device to the network, and select the baud rate of the network.
4. Add the DELT07E1.GSD file to your configuration programs GSD database if it is not already there.
5. Add a **Delta RMC Family** slave device to the network. Refer to Step 2, **Determining the Appropriate GSD Configuration Module,** for details on selecting the correct configuration module.
6. Configure the RMC Slave Device. This involves selecting the correct configuration module for your RMC module and application. Refer to Step 2, **Determining the Appropriate GSD Configuration Module,** for details on selecting the correct configuration module.
7. Select the addresses that the master will use for the registers read from the RMC.
8. Add any other RMC devices you want on the same network. To do this, repeat steps 6 through 8.
9. Save your configuration.
10. Send the configuration to the master device. This step varies greatly depending on the type of master you use.

If you are using *COM PROFIBUS*, *SyCon*, or *SST Profibus Configuration*, you should select one of the following topics for more detailed instructions:

Configuring a PROFIBUS-DP Network with COM PROFIBUS

Configuring a PROFIBUS-DP Network with SST Profibus Configuration

Configuring a PROFIBUS-DP Network with SyCon

See also:

PROFIBUS-DP Overview

10.4.3. PROFIBUS Mode 8-1

This mode has 8 cyclic read registers and 1 explicit Read/Write Register.

Features

This mode has the following features:

- Eight (8) user-selectable cyclic read registers are continuously read for instant access by the PLC or PC.
  See **Configuring the Input Data** below.
- One (1) register anywhere in the RMC70 can be explicitly written or read.
  See **Read from the RMC70** and **Write to the RMC70** below.

- Commands can be issued to any number of axes simultaneously.
  See **Issue a Single Command** and **Issue Simultaneous Commands** below.

Data Blocks

This mode uses two fixed-length blocks on data: the **Command Block** and the **Response Block**.

Command Block

The Command block is a block of 8 contiguous 32-bit output registers. These registers are sent from the PLC or PC to the RMC.

The Command Block has the following structure:

Register Number	Data Type	Description	
0	Integer	Command Register	
		Bit	**Bit Description**
		31	Command Request
		30	Deferred Command
		29	Deferred Command
		20-28	Reserved
		19	Axis 3 Select
		18	Axis 2 Select
		17	Axis 1 Select
		16	Axis 0 Select
		8-15	Reserved
		7-0	Command
1	Float	Command Parameter 0	
2	Float	Command Parameter 1	
3	Float	Command Parameter 2	
4	Float	Command Parameter 3	
5	Float	Command Parameter 4	
6	Integer	Read/Write Register	
		Bit	**Bit Description**
		31	Read/Write
		30	Read/Write Request
		16-29	Reserved
		15-8	R/W Address File
		7-0	R/W Address Element
7	Float*	Write Value	

Registers 0 - 5 are used for issuing commands to the RMC70. See **Issue a Single Command** and **Issue Simultaneous Commands** below for details on using these registers.

Registers 6 and 7 are used for reading and writing to any register in the RMC70. See **Read from the RMC70** and **Write to the RMC70** for details on using these registers.

Response Block

The Response Block is a block of 8 contiguous input registers (cyclic read registers). Each register is a 32-bit word. These registers are continuously sent from the RMC to the PLC or PC. See **Configuring the Input Data** for details on setting up this data.

One of the registers in the Response Block can be set up to return the value of a read from any single register in the RMC. See **Read from the RMC70** below.

The Response Block has the following structure:

Register Number	Data Type	Description
0	Integer	Indirect Data 0 - *must be Axis 0 Status bits!* **Bit** — **Bit Description** 31 — Command Acknowledge 30 — Read/Write Acknowledge 0-29 — Axis 0 Status Bits
1	Float	Indirect Data 1
2	Float	Indirect Data 2
3	Float	Indirect Data 3
4	Float	Indirect Data 4
5	Float	Indirect Data 5
6	Float	Indirect Data 6
7	Float	Indirect Data 7

Using the Data Blocks

Issue a Single Command

Commands are issued by setting up the contents of the first six registers of the Command Block, and when complete, toggling the Command Request bit in the first Command Block register.

To issue a single command to the RMC70, use the following steps:

- Wait until the Command Request bit is equal to the Command Acknowledge bit. If they are not equal, the RMC is currently processing a command request.
- Clear the Deferred Command bits.
- Set the desired Axis select bit in the Command Register. Only one axis can be selected.
- Enter the command in bits 0-7 of the Command Register (0) of the Command Block.
- Toggle the Command Request bit.
- Wait until the Command Request bit is equal to the Command Acknowledge bit. When they are equal, the RMC has received the command.

NOTE:
Until the Command Acknowledge bit matches the Command Request bit, the Input Data registers, including the Status Bits registers do not reflect having received the command.

Example

A Move Absolute (20) command is issued using the PROFIBUS Command Block. Until the Command Request bit matches the Command Acknowledge bit after the Command Request bit has been toggled, the In Position bit should not be checked as it may still be set for the previously requested move. Once the Acknowledge toggles to match, the In Position bit will

have been cleared and when it is set, it is due to the new command being complete. Similar synchronization issues are resolved in the same way with other status bits and registers.

Issue Simultaneous Commands

Although only one command may be sent at a time to the RMC7xP via PROFIBUS, it is possible to simultaneously issue commands to several axes by using deferred commands. Deferred commands are stored in the PROFIBUS command buffer until all deferred commands are received. They are then executed simultaneously. Bits 30 and 29 in the Command Data Register of the Command Block define the deferred status of each command issued. The bits are used as follows:

Bit 30	Bit 29	Action
0	0	**Single Command:** When both bits are zero, the command is not deferred. The command is executed normally. If the PROFIBUS command buffer contains any commands, an error is logged in the Event Log and the commands are removed from the command buffer without being executed. The new command is still issued.
0	1	**Last Deferred:** This command and any deferred commands in the PROFIBUS command buffer are executed simultaneously.
1	0	**First Deferred:** This command is placed as a deferred command in the PROFIBUS command buffer, but is not otherwise processed. If the command buffer already contains commands, an error is logged in the Event Log and the commands are removed from the command buffer without being executed. The new deferred command is still placed in the command buffer.
1	1	**Middle Deferred:** This command is placed as a deferred command in the PROFIBUS command buffer, but is not otherwise processed. This deferred command type allows other deferred commands to be in the command buffer, although they are not required to be there. Note that for a 2-axis controller, this deferred setting will not be used because there can only be a first and last deferred command.

Multiple deferred commands cannot be issued to the same axis. That is, if a deferred command is issued to an axis that already has a deferred command, an error is logged in the Event Log and the previous command is overwritten without being executed.

Configuring the Input Data

The Response Block continuously returns the values from the first eight Indirect Data registers. These registers can be set up to contain the values of any register in the RMC, with the exception of Indirect Data register 0, which *must* be set to contain the Axis 0 Status Bits (register F8:0).

Notice that all the registers in the response block are continuously updated in the PLC or PC. This mode also allows single reads from any single register in the RMC70. See **Read from the RMC70** for details.

Example

The user would like to always return Axis 0 Actual Position and Axis 0 Target Position (registers F8:8 and F8:53). This can be set up in the Indirect Data Map Editor in RMC70Tools. In the Reg# column, enter "F8:8" on the Indirect Data Register 1 row and "F8:53" on the Indirect Data Register 2 row.

Now the Axis 0 Actual Position and Axis 0 Target Position will always be returned in registers 1 and 2 of the Response Block.

Read from the RMC70

The Response Block only returns the values from 8 registers, which must be determined when setting up the communications. However, it is possible to set up one of the registers in the Response Block to return the value of a read from *any* single register in the RMC.

When a read is requested from any single register in the RMC70, the response from this single-register read will be placed in the Axis 0 Read Response register. In order to see the response from the PROFIBUS, you must map the Axis 0 Read Register into one of the Indirect Data Map registers.

Notice that the copy from the requested register into the Axis 0 Read Response register only occurs once, and therefore you will not see the value continuously updating like the other Response Block registers.

To read a register from the RMC, use the following steps:

- Wait until the Read/Write Request bit is equal to the Read/Write Acknowledge bit. If they are not equal, the RMC is currently processing a read or write request.
- Clear the Read/Write bit.
- Set the Read/Write Address File and Read/Write Address Element. See RMC Register Map topic for a description of all RMC registers and their addresses.
- Toggle the Read/Write Request bit.
- Wait until the Read Request bit is equal to the Read/Write Acknowledge bit. When they are equal, the RMC will have updated the Axis 0 Read Response register with the requested data, and the corresponding Response Block register.

To further clarify the ordering, keep these basic rules in mind:

- Do change the read address and Read/Write bit before toggling the Read/Write Request bit.
- Do not change the Read/Write Request bit after a read request until you have processed the data in the Read Response register.
- Do not change the read address or Read/Write bit when the Read/Write Request bit does not match the Read/Write Acknowledge bit.

Write to the RMC70

To write to the RMC, use the following steps:

- Wait until the Read/Write Request bit is equal to the Read/Write Acknowledge bit. If they are not equal, the RMC is currently processing a read or write request.
- Copy the value you wish to write to the RMC into the Write Value register (7) of the Command Block.
- Enter the Read/Write Address file and element. See RMC Register Map topic for a description of all RMC registers and their addresses.
- Set the Read/Write bit.
- Toggle the Read/Write Request bit.
- Wait until the Write Request bit is equal to the Write Acknowledge bit. When they are equal, the RMC has received the data written to it.

To further clarify the ordering, keep these basic rules in mind:

- *Do* change the Read/Write bit, write address, and write value before toggling the Read/Write Request bit.
- Do *not* change the Read/Write bit, write address, or write value when the Read/Write Request bit does not match the Read/Write Acknowledge bit.

See also:

PROFIBUS-DP Overview

PROFIBUS Configuration

10.4.4. PROFIBUS Mode 16-1

This mode has 16 cyclic read registers and 1 explicit Read/Write Register.

This mode is identical to the PROFIBUS Mode 8-1 mode with the addition of a second Response Block of 8 registers. This topic only describes the additional second Response Block. Please read and understand the PROFIBUS Mode 8-1 mode topic before reading this topic.

Features

This mode has the following features:

- Sixteen (16) user-selectable cyclic read registers are continuously read for instant access by the PLC or PC.
- One (1) register anywhere in the RMC70 can be explicitly written or read.
- Commands can be issued to any number of axes simultaneously.

Second Response Block

The second Response Block is a block of 8 contiguous input registers (cyclic read registers), corresponding to the Indirect Data Map registers 8 to 15. All the registers in the second Response Block are configured in the same way as registers 1-7 in the first Response Block in the PROFIBUS Mode 8-1 mode.

The second Response block is non-consistent, that is, each register is not guaranteed to be updated at the same time as the other registers in the block, nor at the same time as the registers in the first Response Block. They are updated at the same rate as the first Response block, but each register's update may differ slightly.

Because it is non-consistent, the second Response Block should not be used for tight synchronization. The following registers should not be placed in Indirect Data Map registers 8 to 15 so that they will not be in the second Response Block:

- Read Response register
- Status and Error bits, because these bits are often checked immediately after a command is issued.

The Second Response Block has the following structure:

Register Number	Data Type	Description
0	Float	Indirect Data 8
1	Float	Indirect Data 9
2	Float	Indirect Data 10
3	Float	Indirect Data 11
4	Float	Indirect Data 12
5	Float	Indirect Data 13
6	Float	Indirect Data 14
7	Float	Indirect Data 15

See also:
PROFIBUS-DP Overview
PROFIBUS Configuration

10.4.5. PROFIBUS Mode 8-7

This mode has 8 cyclic read registers and 7 explicit Read/Write registers.

This mode is identical to the PROFIBUS Mode 8-1 mode with the addition of a Read/Write Block of 8 registers and a Read/Write Response Block of 8 registers. This topic only describes the additional Read/Write Block and Read/Write Response block. Please read and understand the PROFIBUS Mode 8-1 mode topic before reading this topic.

Features

This mode has the following features:

- Eight (8) user-selectable cyclic read registers are continuously read for instant access by the PLC or PC.
- 7 contiguous registers in the RMC70 can be explicitly written or read.
- Commands can be issued to any number of axes simultaneously.

Read / Write Block

The Read/Write Block is a block of 8 contiguous 32-bit output registers. These registers are sent from the PLC or PC to the RMC.

The Read / Write Block has the following structure:

Register Number	Data Type	Description
0	Integer	Read/Write Register **Bit** — **Bit Description** 31 — Read/Write 30 — Read/Write Request 24-29 — Reserved 16-23 — Count 15-8 — R/W Address File 7-0 — R/W Address Element
1	Float	
2	Float	
3	Float	
4	Float	
5	Float	
6	Float	
7	Float	

Register 0 specifies whether to read or write, which register to start reading, and the number of registers.

Registers 1 -7 contain the data to be read from or written to the specified registers.

Read / Write Response Block

The Read/Write Response Block is a block of 8 contiguous input registers (cyclic read registers). Each register is a 32-bit word. These registers are continuously sent from the RMC to the PLC or PC.

The Response Block has the following structure:

Register Number	Data Type	Description
0	Integer	Indirect Data 0 - *must be Axis 0 Status bits!* **Bit** — **Bit Description** 31 — Command Acknowledge 30 — Read/Write Acknowledge 0-29 — Axis 0 Status Bits
1	Float	
2	Float	
3	Float	
4	Float	
5	Float	
6	Float	
7	Float	

See also:

PROFIBUS-DP Overview

PROFIBUS Configuration

10.4.6. Configuring a PROFIBUS-DP Network with COM PROFIBUS

Before reading this topic, you should read and understand PROFIBUS Configuration. This topic only gives a specific example of doing one step of the configuration process. In addition, Siemens may, and probably will, change the steps taken here slightly with each version of *COM PROFIBUS*.

The following steps have been tested with *COM PROFIBUS* versions 3.0 and 3.3:

1. Start *COM PROFIBUS*.
2. If you are modifying an existing PROFIBUS-DP network, open you current configuration file.
3. If you are creating a new PROFIBUS-DP network, you must create a new network and add a master device to the network.
    - On the **File** menu, click **New**.
    - In the **Master & Host Selection** dialog, enter the Address of the master device (in most cases, 1 is a good choice), the Master Station Type, and the appropriate Host Station Type if available. Click **OK**.
    - On the **Configure** menu, click **Bus Parameters**.

- In the **Bus Parameters** dialog, set the **Bus Profile** to **PROFIBUS DP** and the **Baud Rate** to the desired rate. The RMC70P is capable of speeds up to 12000kBaud (12MBaud); check the rates supported by your master and other slaves. Click **OK**.

4. If it is not already in the database, add the DELT07E1.GSD file to your configuration programs GSD database.
    - Copy the DELT07E1.GSD file to the GSD directory under the COM PROFIBUS directory.
    - Copy the DELTRMCN.BMP file to the BITMAPS directory under the COM PROFIBUS directory.
    - On the **File** menu, click **Scan GSD Files**.

5. Add and configure a **Delta RMC Family** slave device to the network.
    - On the **Configure** menu, click **New Slave**.
    - In the **PROFIBUS Address** dialog, select the desired station address, and click **OK**.
    - In the **Family** list, click **Other**.
    - In the **Station Type** list, click **Delta RMC Family**.
    - Click **OK**. In most cases you will be prompted at this time to select a **Preset Configuration**. Refer to the **Determining the Appropriate GSD Configuration Module** section in the PROFIBUS Configuration topic for guidance on selecting the correct option.
    - If you are not prompted to select a **Preset Configuration**, then continue with the following steps:
        - Right-click on the RMC slave device icon, and click **Configure** from the shortcut menu.
        - In the **Configure: Delta RMC Family** dialog, click **Order No.**
        - In the **Select by Order Number** dialog, click the order number determined in the previous steps. Click **Accept**. Click **Close**.
    - Click **OK**.

6. You may wish to assign register addresses within the master at this time.

    This is optional for many masters, because it can be done in the master configuration software. In fact, at times it is not possible to edit the register addresses.

    - Right-click on the RMC slave device icon, and select **Configure** from the shortcut menu.
    - Move the cursor to the top row's **I Addr** cell, and either enter the offset that you wish to access the data at, or click the **Auto Addr.** button. Do the same for the **O Addr** cell.
    - Click **OK**.

7. Add any other RMC devices you want on the same network. To do this, repeat steps 5 and 6.
8. Save your configuration.
9. Send the configuration to the master device. This step varies greatly depending on the type of master you use.

See also:

PROFIBUS-DP Overview

10.4.7. Configuring a PROFIBUS-DP Network with SST Profibus Configuration

Before reading this topic, you should read and understand PROFIBUS Configuration. This topic only gives a specific example of doing one step of the configuration process. In addition, SST may, and probably will, change the steps taken here slightly with each version of *SST Profibus Configuration*.

The following steps have been tested with *SST Profibus Configuration* versions 1.2 and 1.4:

1. Start *SST Profibus Configuration*.
2. If you are modifying an existing PROFIBUS-DP network, open you current configuration file.
3. If you are creating a new PROFIBUS-DP network, you must create a new network and add a master device to the network:
    - On the **File** menu, click **New**.
    - The tree-type control on the left is the GSD library. It contains all devices known about. In this tree, expand **Masters** and **SST** and double-click on the SST master you will be using (e.g. SST-PFB-SLC).
    - In the **Station** box, enter the station address of your master. Click **OK**.
    - On the **Online** menu, click **Network Properties**.
    - In the **Baud Rate** field, select your desired baud rate. The RMC is capable of speeds up to 12000kBaud (12MBps); check the rates supported by your master and other slave devices. Click **OK**.

4. Add the DELT07E1.GSD file to your configuration programs GSD database.
    - In the GSD library tree, expand **Slaves** and **Delta Computer Systems, Inc.** and right-click on the **Delta RMC Family** entry. On the shortcut menu, click **Delete**. Click **Yes** when asked.
    - Click **New Device**, which is the left-most button on the GSD library toolbar.
    - Browse to the DELT07E1.GSD that shipped with RMC70Tools, select that file, and click **Open**.

5. Add and configure a **Delta RMC Family** slave device to the network.
    - In the GSD library tree, expand **Slaves** and **Delta Computer Systems, Inc.** and double-click on **Delta RMC Family**.
    - On the **General** tab, select the station address of your master in the **Station** field of the dialog that is displayed.
    - In the **Modules** tab, click **Add**. In the dialog that is displayed, select the appropriate module from the **Available Modules** list and click **OK**. For help on determining the appropriate configuration module, refer to the **Determining the Appropriate GSD Configuration Module** section of the PROFIBUS Configuration topic.
    - In the **Address** (or **SLC Address**) tab, review the default addressing and change it if necessary.
    - Click **OK**.

6. Add any other RMC devices you want on the same network. To do this, repeat step 5.
7. Save your configuration.
8. Send the configuration to the master device. This step varies depending on the master you selected.

See also:
PROFIBUS-DP Overview

10.4.8. Configuring a PROFIBUS-DP Network with SyCon

Before reading this topic, you should read and understand PROFIBUS Configuration. This topic only gives a specific example of doing one step of the configuration process. In addition, Hilscher may, and probably will, change the steps taken here slightly with each version of *SyCon System Configurator*.

The steps that follow came from *SyCon System Configurator 2.082*:

1. Start *SyCon System Configurator*.
2. If you are modifying an existing PROFIBUS-DP network, open you current configuration file.
3. If you are creating a new PROFIBUS-DP network, you must create a new network and add a master device to the network.
    - On the **File** menu, click **New**. If you have multiple networks installed you will need to then select the network type: select PROFIBUS. If you are not given the option of selecting PROFIBUS, you may not have installed the PROFIBUS driver for SyCon.
    - On the **Insert** menu, click **Master**. Move the cursor to the top device slot in the window area (the cursor will change to an arrow with an M next to it), and click to place the master device.
    - In the **Insert Master** dialog, select the desired master and click the **Add>>** button. Enter the station address, and click **OK**.
    - On the **Settings** menu, click **Bus Parameter**.
    - In the **Bus Parameter** dialog, set the **Baud Rate** to the desired rate. The RMC is capable of speeds up to 12000kBits/s (12MBaud); check the rates supported by your master and other slaves. Click **OK**.
4. Add the DELT07E1.GSD file to your configuration programs GSD database.
    - You must first have an open PROFIBUS project file.
    - On the **File** menu, click **Copy GSD**.
    - In the **Open** dialog, navigate to the directory where the DELT07E1.GSD file is located (by default it is installed in the main RMC70Tools directory), select that file and click **Open**.
    - **NOTE:** This will not add the bitmap file. If you wish to use the bitmap file; copy the DELTRMCN.BMP file to the Fieldbus\Profibus\BMP directory under the SyCon directory (by default, the total path will be C:\Program Files\Hilscher GmbH\SyCon\Fieldbus\Profibus\BMP).

5. Add a Delta RMC70 Family slave device to the network.
    - On the **Insert** menu, click **Slave**. Move the cursor to the next available device slot in the window area (the cursor will change to an arrow with an *S* next to it), and click to place the device.
    - In the **Insert Slave** dialog, click **Delta RMC Family**, and then click the **Add>>** button. Enter the station address, and click **OK**.

6. Configure the RMC70 Slave Device.

    This involves selecting the correct configuration module for your RMC module and application. Refer to the sub-topics **Determining the Appropriate GSD Configuration Module** in the PROFIBUS Configuration topic for details on selecting the correct configuration module.
    - Right-click on the RMC slave device icon, and click **Slave Configuration** from the shortcut menu.
    - In the **Slave Configuration** dialog, from the list of order numbers, select the configuration module you selected in the previous steps.
    - Click **Append Module**.
    - You may change the **I Addr.** and **O Addr.** fields for the added module to set the offsets of the data.
    - Click **OK**.

7. Add any other RMC70 devices you want on the same network. To do this, repeat steps 5 and 6.
8. Save your configuration.
9. Send the configuration to the master device. This step varies depending on the master you selected.

See also:

PROFIBUS-DP Overview

10.5. Serial (RS-232/485)

10.5.1. Serial Communications Overview

The RMC7*x*S CPU modules provide serial communications, allowing the RMC70 to interface to other devices such as HMIs and PLCs. The RMC7*x*S module is passive in that it always waits for a request from a master serial device before responding. It does not initiate communications.

Features

The RMC7*x*S serial port has a great amount of flexibility, as summarized below:

- **Protocols:** Allen-Bradley DF1 (Full- and Half-duplex), Modbus/RTU
- **Baud Rate:** 9,600 to 115,200
- **Parity:** Odd, Even, or None
- **Line Drivers:** RS-232 and RS-485
- **Termination and Biasing (RS-485 only):** Both jumper-selectable on the connector

Setup

Because of the large number of options offered by the RMC70, setting it up can be intimidating to users new to serial communications. Read each of the following topics carefully before designing your serial network:

Configuring the RMC7*x*S Serial Communications

Line Drivers: RS-232/485

Serial Network Topologies

RS-232 Wiring for the RMC70

RS-485 Wiring for the RMC70

RS-485 Termination and Biasing

Protocols

The following topics describe how to use each protocol supported by the RMC7*x*S:

DF1 (Full- and Half-Duplex)

Modbus/RTU

Using Serial Communications

Issuing Commands

Issuing Commands with Serial Communications

This topic describes how to issue commands to the RMC70 via serial communication.

Command Area Registers

The following registers are called the Command Area registers. You must write to these registers when issuing commands.

Element	Register Name
F16:0	Axis 0 Command
F16:1	Axis 0 Parameter 1
F16:2	Axis 0 Parameter 2
F16:3	Axis 0 Parameter 3
F16:4	Axis 0 Parameter 4
F16:5	Axis 0 Parameter 5
F16:6-11	Axis 1 Command Registers
F16:12-17	Axis 2 Command Registers
F16:18-23	Axis 3 Command Registers

Issuing a command

How you issue commands depends on how may registers your system can write at once to the RMC70.

One Register at a Time

If your system can only write one register at a time to the RMC70, it is important that you write to the Command Area registers in the correct order.

To issue a command to an axis:

1. Write values to the Command Parameters first. You need only write to the Command Parameters used by the command you will issue.
2. Write the number of the command to the Command Register.

You cannot issue simultaneous commands directly using this method. You can issue a command simultaneously to multiple axes from a User Program.

Multiple Registers at a Time

Issuing commands is simpler using this method.

To issue a command:

1. Write to the Command Register and all the command Parameters in the same write.

If your system can write multiple registers at a time to the RMC70, you can write to multiple axes simultaneously and thereby issue commands simultaneously.

10.5.2. Configuration and Wiring

10.5.2.1. Line Drivers: RS-232/485

The RMC7*x*S module supports two different line drivers: RS-232 and RS-485 (2-wire only). See the Configuring Serial Communications topic for instructions on how to choose a line driver.

Features

The following chart compares these line drivers as implemented in the RMC70:

	RS-232	RS-485(2-wire)
Duplex	Full	Half
Differential?	No	Yes
Topology	Point-to-point	Point-to-point, Multi-drop
Wires	3	2 + CMN
Max Length[2]	50-100 ft	4000 ft

Note:
The maximum cable lengths vary depending on the baud rate, termination (for RS-485), and capacitance of the cable. See the RS-485 Termination and Biasing topic for details.

Each of the above features is described below:

Duplex (Full or Half)

Full-duplex means that each device on a serial network can send and receive at the same time, effectively doubling the bandwidth of the network. Half-duplex means that only one device on the network can send data at one time. For the above drivers, full-duplex requires separate send and receive wires.

Differential

Differential wiring uses two wires per signal which allows common mode noise rejection. RS-232 does not use differential wiring, but instead has one wire per signal plus a ground. Differential wiring allows for longer cable distances and greater noise immunity.

Topology

Topology describes the layout of the network. Point-to-point means that exactly two devices are wired together. Multi-drop means that two or more devices are chained together. Notice that "multi-drop" with only two devices becomes point-to-point. See Serial Network Topologies for details.

Wires

This item refers to how many wires need to be connected between nodes. Notice that 2-wire actually requires three because of the Common in addition to the differential signal wires. See RS-232 Wiring for the RMC SERIAL and RS-485 Wiring for the RMC SERIAL for details.

Max Length

The maximum cable lengths vary depending on the baud rate, termination (for RS-485), and capacitance of the cable. See the RS-232 Wiring for the RMC70 and RS-485 Wiring for the RMC70 topics for details.

Flow Control

Flow control can be used with RS-232 to ensure that one device does not overrun the other device. That is, if one device is sending data and the receiving device's buffers get full, then it can use flow control to pause the first device's sending until it has room in its buffers.

10.5.2.2. Configuring the RMC7xS Serial Communications

The RMC7*x*S supports most standard serial port options, several protocols, and several line drivers. These settings can be changed using RMC70Tools and saved to Flash memory.

Changing Serial Settings

To change the serial communication settings:

1. In the **Project** pane, right-click the desired controller and click **Properties** on the shortcut menu.
2. Click the **Serial** tab.
3. Make the desired changes. For a description of the options, see the **Configuration Options** section in this topic.
4. Click **OK**.

Configuration Options

The following options are available under the **Serial** tab in the controller **Properties** dialog:

Serial Port

Select which port on the RMC7xS to communicate through. The ports are labeled on the front panel of the RMC7*x*S.

Note:
The **0-RS232 Monitor** port is used by RMC70Tools, but can be used for other serial communication as well.

Protocol Settings

Select the protocol supported by your master device and the address of the RMC7*x*S. The address must match the address that master device expects.

Serial Port Settings

The following settings define how data is sent over the wire:

- **Line Driver:** Select the line driver you wish to use. For further details on the line drivers, see Line Drivers: RS-232/485.
- **Baud Rate:** Select the baud rate from 9,600 to 115,200. This must match the other device(s) on the network.

NOTE:
Some of these options may be disabled depending on which protocol was selected. For example, many protocols require eight data bits, and as such seven data bits is not available when these protocols are selected.

Advanced...

These are advanced settings for the line drivers:

- **Data Length:** Select seven (7) or eight (8) data bits. Most protocols require eight data bits. This must match the other device(s) on the network.
- **Parity:** Select which type of parity error checking you want to include. This must match the other device(s) on the network.
- **Stop Bits:** Select the number of stop bits. This must match the other device(s) on the network.

10.5.2.3. Serial Network Topologies

The RMC7*x*S supports two network topologies: point-to-point and multi-drop. Which topologies are available depend on the line driver (RS-232 or RS-485) used. See Line Drivers: RS-232/485 for details on choosing the appropriate line driver.

Point-to-Point

Point-to-point means that exactly two devices are wired together. For the RMC, this means that there will be one RMC wired to one host. Both line drivers support point-to-point, as shown in Figures 1 and 2:

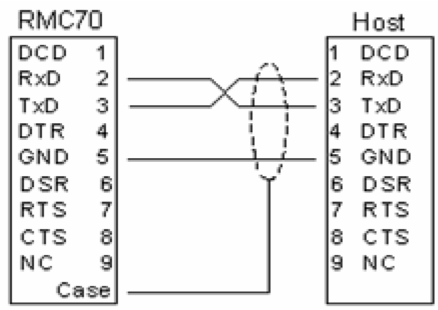


Fig. 1: Point-to-Point RS-232 Network

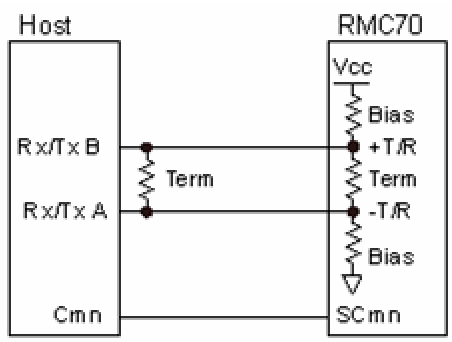


Fig. 2: Point-to-Point RS-485 Network

Figure 2 shows biasing and termination. Termination and biasing can be left out of networks at the expense of maximum cable distance and noise immunity. See RS-485 Termination and Biasing for details.

Multi-Drop

Only RS-485 supports multi-drop. Multi-drop is the connecting of multiple slaves with a single master. Slaves should be chained together. Neither a star topology nor a chain with long stubs (wires from the main chain to the device) should be used. These topologies will cause excessive ringing on the network and unreliable data transmission.

The number of devices that can be connected to the network is dictated by the number of unit loads that each represents. According to the TIA/EIA-485-A specification, there can be a maximum of 32 unit loads connected to a single network. Each RMC represents unit load for a total of 124 RMCs on the network, assuming the host is a unit load.

Figure 3 shows a typical multi-drop chain:

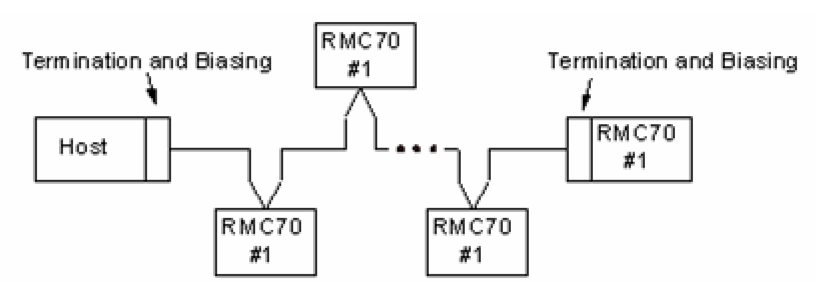


Fig. 3: Daisy-Chained RS-485 Network

Note: Termination should only be located at the extreme ends of the network:

Figure 4 shows one host with two RMC controllers in a daisy-chained two-wire RS-485 configuration:

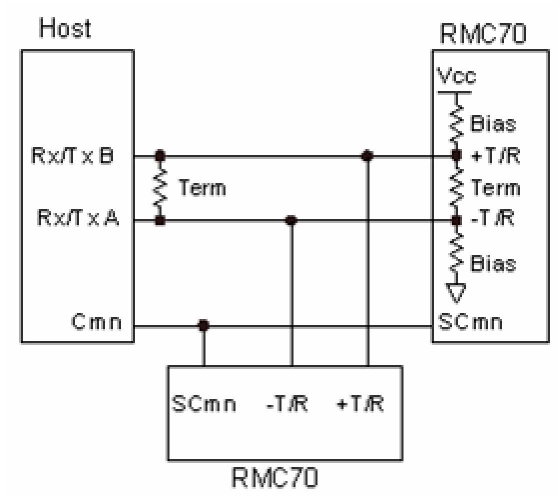


Fig. 4: Two-Wire Multi-drop RS-485 Network

10.5.2.4. RS-232 Wiring for the RMC70

Connectors

Both of the 9-pin male DB connectors on the RMC7xS are used for RS-232 communications. A 9-pin female connector is required on the RMC70 end of the connecting cable. The other end of the cable must match the master/host hardware, which may be a 9-pin, 25-pin, RJ-11, connector block, or other connector.

See also Wiring Guidelines.

Pin-Out

Pin assignments on the RS-232 9-pin connector:

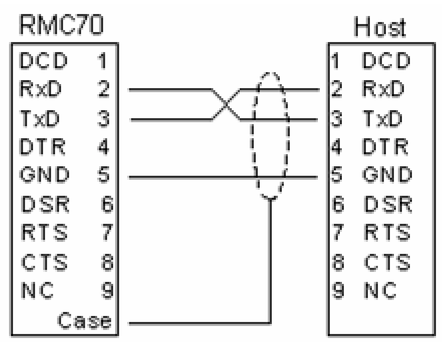
Pin #	RS-232 Function
1	DCD- Not used by RMC
2	RxD - Receive Data
3	TxD - Transmit Data
4	DTR - Not used by RMC
5	GND - Common
6	DSR - Not used by RMC
7	RTS - Not used by RMC
8	CTS - Not used by RMC
9	Not Connected

Cabling

A null-modem or crossover cable is typically used for RS-232 communications. The RMC70 RS-232 communications require only three conductors in the cable: RxD, TxD, and GND. See the following wiring diagram for details.

Delta recommends that a shielded cable be used to limit susceptibility to outside electrical interference.

Cable Wiring



Cable Length

One of the characteristics that limit the length of an RS-232 cable is capacitance. Most cables have a capacitance rating in pF/ft. Use the following formula to calculate the maximum distance signals can be reliably transmitted for a given cable capacitance.

MaxLength = 2400 pF / (C + (Shield * C)),

where

Shield	=	2 for shielded cable and 0.5 for unshielded cable
C	=	capacitance rating of the cable in pF/ft
2400 pF	=	derived by taking the maximum capacitance specified by ANSI/TIA/EIA-232-F (2500 pF) and subtracting 100 pF for the input capacitance of the receiver

Example:

The cable that is being used is shielded and has a capacitance of 15 pF/ft.

MaxLength	=	2400 pF / (15 pF/ft + (2 * 15 pF/ft))
	=	2400 pF / (45 pF/ft)
	=	53.3 ft

Note:
If you need to run your communications farther than allowed by this formula, then you must either use a cable with lower capacitance or use RS-485.

10.5.2.5. RS-485 Wiring for the RMC70

Connectors

The RMC70 supports only 2-wire RS-485. RS-485 uses 8-pin connector block on the RMC75S CPU module (pins 6, 7 and 8 are for power to the controller). The pin-out is as follows:

Pin	RMC70 Label	RS-485 (2-wire) Function
1	+T/R	Rx/Tx B (+)
2	Trm Jpr	See RS-485 Termination and Biasing
3	-T/R	Rx/Tx A (-)
4	SCmn	Common
5	Bias Jumper	See RS-485 Termination and Biasing

NOTE:
Pins 2 and 4 are for termination and biasing. See the RS-485 Termination and Biasing topic for details.

NOTE:
Some manufacturers use A and B labeling, while others use + and - labeling. If you need to interface to equipment that uses an alternate labeling scheme, keep in mind that A corresponds to - and B corresponds to +.

See also Wiring Guidelines.

Cabling

All cabling for balanced or differential communications should consist of twisted pairs. Because the RMC's RS-485 interface is isolated, the signal common must be run alongside or in the cable. Therefore, for a two-wire network the cable must be either a one-pair cable with a separate ground line that is run externally or a two-pair cable in which one pair is used as the common. For a clean cabling solution, Delta recommends the option using an additional wire pair.

Another consideration when selecting communication cabling is the impedance of the cable. This impedance should match the termination resistance that is used. See RS-485 Termination and Biasing to determine whether or not your network will require termination.

No cable characteristics are specified in the TIA/EIA-422-B and TIA/EIA-485-A standards, but the RS-422-B standard does recommend 24AWG twisted pair cable with capacitance of 16 pF/ft and 100Ω characteristic impedance. These specifications will work well for RS-485 as well. One good choice would be to use Category 5 Ethernet cable. Category 5 Ethernet cable has a capacitance of 17 pF/ft max with 100Ω characteristic impedance. It is commonly available as shielded twisted pair (STP) or unshielded twisted pair (UTP). If this is not suitable then there are a number of manufacturers of communications cable such as Alpha and Beldon Wire and Cable.

Cable Length

The maximum cable length for RS-485 depends on the baud rate and termination. At higher baud rates, termination allows longer cable lengths. For details on the effects of termination and how to apply it, see RS-485 Termination and Biasing.

The following chart shows the maximum cable length for RS-485 with and without termination:

Maximum RS-485 Cable Length:

Baud Rate	Max Unterminated Cable Length (ft)	Max Terminated Cable Length (ft)
115,200	475	3250

57,600	950	4000
38,400	1900	4000
19,200	3750	4000
9,600	4000	4000
4,800	4000	4000
2,400	4000	4000

10.5.2.6. RS-485 Termination and Biasing

Termination and Biasing are concepts that only apply to differential wiring. As such, they only apply to RS-485 and not RS-232.

Note:
Delta recommends you always use biasing on a RS-485 network. To determine whether you need termination, read the **Termination Concept** section in this topic.

Selecting Termination and Biasing on the RMC70

The RMC7xS termination and biasing can be independently selected with jumpers on the 8-pin RS-485 connector. The locations where the jumpers should be installed are marked on the label.

To select termination: Insert a jumper between pins **+T/R** and **Trm Jpr**.

To select biasing: Insert a jumper between pins **SCmn** and **Bias Jumper**.

The Termination Concept

Cable termination is a way of absorbing transmitted energy at the end of a network. This prevents signal reflections from bouncing back towards the transmitter and potentially upsetting signal quality and communications.

The termination resistor should match the characteristic impedance of the cable being terminated. The effective impedance of the RMC7*x*S's termination resistor and biasing resistors is 114Ω. Therefore, cabling with impedance of 100Ω to 120Ω is recommended.

Termination should be placed at the end of the network for each wire pair. For RS-485 (2-wire, point-to-point or multi-drop), terminate the wire pair at each end of the network. The diagram in Serial Network Topologies shows the correct location of the termination.

Termination and Cable Length

Termination is not required on all differential networks, but it does typically extend the maximum cable length. The following chart shows the maximum cable lengths at various baud rates with and without termination:

NOTE:
The maximum cable length is the length of the entire network and not just the distance between nodes on the network.

Termination vs. Cable Length:

Baud Rate	Max Unterminated Cable Length (ft)	Termination Requirements	Max Terminated Cable Length (ft)
115,200	475	Required beyond 475 ft	3250
57,600	950	Required beyond 950 ft	4000
38,400	1900	Required beyond 1900 ft	4000
19,200	3750	Required beyond 3750	4000

		ft	
9,600	4000	Not Required	4000
4,800	4000	Not Required	4000
2,400	4000	Not Required	4000

Cable Length Derivation

The values presented in the chart above are based on 24AWG cable with capacitance of 16 pF/ft and the following reasoning. Signals travel through a cable at approximately 66% of c or 0.66 ft/ns. It is assumed that a signal transition will dampen out after three round trips in the cable. This damping must occur before the bit is sampled or within half a bit time. One bit time is equal to the reciprocal of the baud rate.

Example:

Compute the cable length for 115,200 baud RS422.

First, we compute a half bit time at this baud rate.

Half Bit Time = 0.5 * 1 / 115200
= 4,340 ns

Next, we convert this time to the distance the signal would travel in this time, assuming a speed of 0.66 ft/ns as described above:

Distance = 4,340 ns * 0.66 ft / ns
= 2890 ft

Since three round trips are required for the signal transition to dampen and each round trip is twice the length of the cable, the total distance in feet is divided by six to get the final unterminated cable length:

Length = 2890 ft / 6
= 482 ft

This value is then rounded down to allow for inexact cable velocities and damping rates, giving us 475 ft.

The Biasing Concept

RS-485 indicates a binary 1 when the A line is at least 200 mV negative with respect to B, and a binary 0 when A is at least 200 mV positive with respect to B. It is important that the lines always be in a known state, not only when being driven. Biasing forces the network into a known state when the lines are idle and therefore otherwise not driven.

A known state is forced by allowing current to flow across the termination resistor. Therefore, biasing is usually selected on the RMC that also has a termination. However, some masters only have termination, in which case the user may want to only select biasing on an RMC close to the master. The current will then flow across the master's termination resistor.

The RMC7*x*S requires biasing in order to be in a known state when the lines are idle. The biasing forces a binary 1.

Example:

This example assumes that there is a single master and two RMCs on the network. Compute the voltage across a 120Ω termination resistor when using 1150Ω biasing resistors.

First, we calculate how much DC resistance will be between the biasing resistors. Calculating the parallel resistance of all DC terminations and node input impedances does this. For a single master and two RMCs we have the following components:

- Master load: 1 unit load, which is defined as 12 kΩ.
- RMC loads: unit load each, which is 48 kΩ.
- Termination Resister in the RMC: 120Ω

Therefore, putting all of the resistances in parallel yields the following:

$$\text{Termination Resistance} = 120\Omega \,||\, 12k\Omega \,||\, 48k\Omega \,||\, 48k\Omega$$
$$= 118\Omega$$

Then, we calculate how much DC resistance the network has between power rails:

$$\text{Total Resistance} = 1150\Omega + 118\Omega + 1150\Omega$$
$$= 2418\Omega$$

Next, we calculate how much current is flowing through this DC resistance:

$$\text{Current} = 5\text{VDC} / 2418\Omega$$
$$= 2.068\text{mA}$$

Finally, we calculate the voltage drop across the termination resistor:

$$\text{Voltage} = 2.068\text{mA} * 118\Omega$$
$$= 244\text{mV}$$

This value is greater than the 200mV difference required by the TIA/EIA standards and constitutes a valid binary 0 state.

10.5.3. RMC70 Serial Protocols

10.5.3.1. DF1 Protocol (Full-and Half-Duplex)

DF1 is one of the serial protocols supported by the RMC70S. The RMC70 also communicates to RMC70Tools via DF1. This topic describes the DF1 protocol as it applies to the RMC70. For details on configuring the RMC70 for serial communications, including DF1, see the Serial RMC70S Configuration topic.

General

Allen-Bradleys DF1 Protocol and Command Set Reference Manual (pub. no. 1770-6.5.16) is the authority on the DF1 full- and half-duplex protocols. This manual is available on Allen-Bradleys web site (http://www.ab.com). As of this writing, the following URL contains this document: http://www.ab.com/manuals/cn/17706516.pdf. When this link is out-of-date, try searching for the above publication number.

Full-duplex DF1 is used for peer-to-peer communication. Therefore, only two devices can communicate with one another. The RMC70 only supports full-duplex when used with RS-232.

Half-duplex DF1 is used for master-slave communication with one or more slaves. When more than two devices communicate with one another, 2-wire RS-485 is used. Otherwise, any line driver can be used.

RMC70 Support for DF1

Both full- and half-duplex DF1 use the same application protocol, which consists of commands and functions for the slave or peer to execute.

The RMC70 supports the following DF1 functions:

- PLC5 Word Range Write (CMD=0x0F, FNC=0x00)
- PLC5 Word Range Read (CMD=0x0F, FNC=0x01)
- PLC5 Typed Write (CMD=0x0F, FNC=0x67)
- PLC5 Typed Read (CMD=0x0F, FNC=0x68)
- SLC Protected Typed Write with 2 Address Fields (CMD=0x0F, FNC=0xA9)
- SLC Protected Typed Read with 2 Address Fields (CMD=0x0F, FNC=0xA1)
- SLC Protected Typed Write with 3 Address Fields (CMD=0x0F, FNC=0xAA)
- SLC Protected Typed Read with 3 Address Fields (CMD=0x0F, FNC=0xA2)
- Echo (CMD=0x06, FNC=0x00)
- Diagnostic Status (CMD=0x06, FNC=0x03)

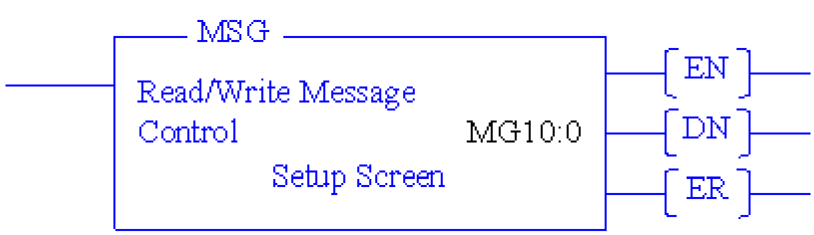
PLC Support for DF1

DF1 is a major industrial serial protocol supported by a large number of devices, both those built by Allen-Bradley and other companies. Any DF1 master implementation that uses the above blocks should also be able to read and write from the RMC70.

Each Allen-Bradley PLC uses the Message (MSG) ladder logic block to initiate reads and writes over a serial port. For full details on this block, refer to Allen-Bradleys Instruction Set Reference Manual for the appropriate PLC. While the same block is used by each PLC, the semantics differ slightly for each. Below are specific instructions for the PLC-5, SLC 5/0x, and MicroLogix. This documentation came using RSLogix 5 version 3.2.0.0 and RSLogix 500 version 4.10.00.

Allen-Bradley PLC-5

The PLC-5 MSG block is displayed as follows:



PLC-5 MSG Block Parameters

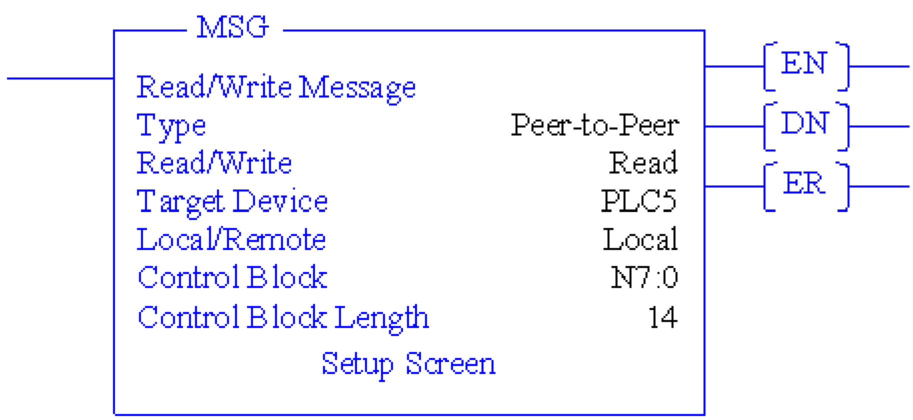
Control
This parameter points to a message (MG) file type element or a block of integer (N) file type elements. The number of N-file elements required varies from 11 to 15; look at the BLOCK SIZE field in the setup screen for the exact size. Set this to an unused block of registers, and then use the Setup Screen option in the MSG ladder logic block to modify those register values:

- **This PLC-5:** This section holds parameters for the PLC-5.
  - **Communication Command:** From this drop-down list, select PLC-5 Typed Read to read values from the RMC70, or PLC-5 Typed Write to write values to the RMC70.

    - **Data Table Address:** Enter the address of the first Allen-Bradley PLC register to read RMC70 registers into, or to write to RMC70 registers from.
    - **Size in Elements:** Enter the number of RMC70 registers to read or write in this field. Transfers are limited to 1000 bytes for PLC-5 Typed Reads and Writes. Therefore, this limit is 500 integers, 250 floats, etc. Notice that this limit is larger than the number of elements in the RMC70's N-files. Reads or writes that extend beyond the end of a register file will continue into the next register file. For example, reading 300 elements from N9:0 will read N9:0 to N9:255, then N10:0 to N10:43.
    - **Port Number:** Set this to the channel number of the serial port you want to use.
- **Target Device:** This section holds parameters for the target device.
    - **Data Table Address:** Enter the address of the first RMC70 register to read or write in this field.
    - **MultiHop:** This parameter should be set to **No**.
    - **Local Node Addr (dec):** Enter the node address of this RMC70. The node address of the RMC70 is set up in the Serial Module Options dialog, which is described in Configuring the RMC70 SERIAL. The node address entered on that screen is in decimal, so it is recommended that you enter the same number in the dec field.

Allen-Bradley SLC 5/0x

The SLC 5/05 MSG block is displayed as the following:



SLC-5/05 MSG Block Parameters

Type

This parameter is always set to Peer-to-Peer for serial communication channels.

Read/Write

This parameter should be set to Read to read registers from the RMC70, and to Write to write registers to the RMC70.

Target Device

This parameter has possible values of 500CPU, 485CIF, and PLC5. This should be set to PLC5 or 500CPU for communicating with the RMC70.

Local/Remote

This parameter has possible values of Local and Remote. This will be set to Local when the RMC70 is on the same serial network as the SLC.

Control Block

This parameter points to a block of 12 integer-file registers. Set this to a block of

registers, and then use the Setup Screen option in the MSG ladder logic block to modify those register values:

- **This Controller:** This section holds parameters for the SLC 5/05.
    - **Communication Command:** This parameter will be set to PLC5 Read, PLC5 Write, 500CPU Read, or 500CPU Write, depending on what was selected in the MSG block itself (as described above).
    - **Data Table Address:** Enter the address of the first Allen-Bradley PLC register to read RMC70 registers into, or to write to RMC70 registers from.
    - **Size in Elements:** Enter the number of RMC70 registers to read or write in this field. The range enforced by the SLC depends on the model used. The SLC 5/02 can transfer 1 to 41 integers, the SLC 5/03 and 5/04 can transfer 1 to 103 integers, and the SLC 5/05 can transfer 1 to 256 integers. Reads or writes that extend beyond the end of a register file will continue into the next register file. For example, reading 256 elements from N9:128 will read N9:128 to N9:255, then N10:0 to N10:127.
    - **Channel:** Set this to the channel number of the serial port you want to use. Keep in mind that on the SLC 5/05 channel #1 is the Ethernet channel.
- **Target Device:** This section holds parameters for the target device.
    - **Communication Command:** From this drop-down list, select PLC-5 Typed Read to read values from the RMC70, or PLC-5
    - **Message Timeout:** Indicate the number of seconds to wait for the RMC70 to respond before determining that the attempt failed. This can be set as low as a few seconds.
    - **Data Table Address:** Enter the address of the first RMC70 register to read or write in this field.
    - **Local Node Addr (dec):** Enter the node address of this RMC70. The node address of the RMC70 is set up in the Serial Module Options dialog, which is described in Configuring the RMC70 SERIAL. The node address entered on that screen is in decimal, so it is recommended that you enter the same number in the dec field.
    - **Local/Remote and Bridge Parameters:** In most applications these will be set to Local and there will be no bridge parameters. If you are using a bridge then these parameters will need to be used. However, this is beyond the scope of RMC70 documentation. Refer to your Allen-Bradley documentation for instructions on using these fields.

Allen-Bradley MicroLogix

The MicroLogix MSG block is displayed as follows:

To edit the parameters of the message block, select the MSG block, enter an unused MSG file in the MSG File parameter, and double-click Setup Screen. This brings up a dialog with the following options:

This Controller: This section holds parameters for the MicroLogix.

Communication Command: This parameter can be set to PLC5 Read, PLC5 Write, 500CPU Read, or 500CPU Write. The type of PLC selected is not important, but the Read or Write determines whether you will read registers from the RMC70 or write registers into the RMC70.

Data Table Address: Enter the address of the first Allen-Bradley PLC register to read RMC70 registers into, or to write to RMC70 registers from.

Size in Elements: Enter the number of RMC70 registers to read or write in this field. The MicroLogix can transfer 1 to 41 integers.

Channel: Set this to the channel number of the serial port you want to use. The MicroLogix 1500 LRP Series B can use either channel 0 or 1, but all other MicroLogix PLCs will always use channel 0.

Target Device: This section holds parameters for the target device.

Message Timeout: Indicate the number of seconds to wait for the RMC70 to respond before determining that the attempt failed. This can be set as low as a few seconds.

Data Table Address: Enter the address of the first RMC70 register to read or write in this field. See the RMC70 Register Map (Allen-Bradley) for help on addresses.

Local Node Addr (dec): Enter the node address of this RMC70. The node address of the RMC70 is set up in the Serial Module Options dialog box, which is described in Configuring the RMC70 SERIAL. The node address entered on that screen is in decimal, so it is recommended that you enter the same number in the dec field.

Local/Remote and Bridge Parameters: In most applications these will be set to Local and there will be no bridge parameters. If you are using a bridge then these parameters will need to be used. However, this is beyond the scope of RMC70 documentation. Refer to your Allen-Bradley documentation for instructions on using these fields.

10.5.3.2. Modbus/RTU Protocol

Modbus/RTU is one of the serial protocols supported by the RMC70. This topic describes the Modbus/RTU protocol as it applies to the RMC70. For details on configuring the RMC70 for serial communications, see the Serial Configuration topic.

General

Modbus/RTU is a standard protocol managed by Schneider Automation (Modicon). The specification is available through Schneider Automation (http://www.modicon.com/openmbus) and Modbus.org (http://www.modbus.org). The Modicon Modbus Protocol Reference Guide (PI-MBUS-300) from Modicon is a good reference for this protocol. It is available through the Modicon web site. At the time of this writing it was necessary to select the MODBUS/TCP Protocol documents link to find an Acrobat Reader (.pdf) format version of this document.

RMC70 Support for Modbus/RTU

Modbus, in its various forms such as Modbus/ASCII, Modbus/RTU, Modbus Plus, and Modbus/TCP, is a request/response protocol. That is, a Modbus master makes a request from a Modbus slave, and the slave responds. A number of functions are defined under Modbus. The following functions are supported by the RMC70:

- 03: Read Holding Registers
- 06: Preset Single Register
- 16 (10 Hex): Preset Multiple Registers
- 23 (17 Hex): Read/Write 4X Registers

Each of the above functions acts on 4X or Holding registers. These 4X registers are mapped in the RMC70 as described in the following topic:

- RMC Register Map (Modbus/TCP)

11.Command Reference

11.1. RMC70 Commands Overview

The RMC70 has a rich set of pre-programmed commands that perform anything from simple moves to complex motions to system control. Each command is represented by an integer number in floating point format. For a list of all the RCM70 commands, see the Command List topic.

Issuing Commands

Commands can be issued to the RCM70 in the following ways:

- Using the Command Tool in RMC70Tools.
- From a User Program.
- From a PLC (or other host controller).
  See the specific communication type for details.

Command Format

A command consists of a command number followed optionally by one to five command parameters. In RMC70Tools, the commands are always referred to with a descriptive name and a number. When communicating with the RMC from a host system, commands are referred to by their number.

In the RMC70 register map, the command consists of one block of six 32-bit command registers per axis.

Register	Description
0	Command Number. Each command is represented by an integer in 32-bit floating point format. **NOTE:** The high 24 bits of this register are reserved. The command number is limited to the low 8 bits.
1	Parameter 1 - Primary command parameter
2	Parameter 2 - Additional parameter
3	Parameter 3 - Additional parameter
4	Parameter 4 - Additional parameter
5	Parameter 5 - Additional parameter

Command List

For a list of commands, see the Command List topic.

11.2. RMC70 Commands

This is a complete list of the commands currently available on the RMC, grouped by type. Each command is represented by an integer number in floating point format. The command number is given in parentheses. Click on a command for a description. For general command information, see the Command topic.

General

- No-op (0)
- Clear Faults (4)
- Initialize Axis (7)
- Fault Controller (8)
- RUN Mode (98)
- PROGRAM Mode (99)

Motion

Open Loop

- Direct Drive (9)
- Open Loop Rate (10)
- Open Loop Absolute (11)
- Open Loop Relative (12)

Point-to-Point

- Move Absolute (20)
- Move Relative (21)
- Quick Move Absolute (15)
- Quick Move Relative (16)
- Time Move Absolute (23)
- Time Move Relative (24)

Stops

- Closed Loop Halt (1)
- Open Loop Halt (2)
- Disable Drive (3)
- Hold Current Position (5)
- Stop (6)

Specialty

- Speed at Position (36)

System

- Update Flash (110)

Pressure Control

- Enable/Disable Pressure Limit (40)
- Ramp Pressure (41)

Set Parameters

- Set Integrator Mode (71)
- Set Enable Output (67)
- Set Actual Position (49)
- Set Target Position (48)
- Read Register (111)
- Write Register (112)

Programming

- Stop Task (91)
- Start Task (90)

Plots

- Enable/Disable Plot Trigger (104)
- Rearm Plot (103)
- Start Plot (100)
- Stop Plot (101)
- Trigger Plot (102)

Step Editor Commands

- Expression (113)

11.3. General Commands

11.3.1. RMC70 Command: No-op (0)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This is a space filler command that can be sent to the RMC70 when no action is expected from the controller. It is used in the following cases:

- On steps in the Step Table that have no command.
- To issue a command to only one axis when using the Command Tool, you must issue the No-op (0) command to the other axis.

Command Parameters

None.

11.3.2. RMC70 Command: Clear Faults (4)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command clears all latched error bits, unless the underlying condition still exists. It also clears the Halted and External Halt status bits unless error bits remain set that would otherwise set these bits.

Except for its effect on the error bits and halt status bits, this command does not affect the state of the axis. That is, it will not stop a halting axis from continuing to ramp down its Control Output, and it will not switch an axis into or out of closed loop.

Command Parameters

None.

Typical Uses

This command is often unnecessary because many commands clear the above-mentioned status and error bits. However, it may be useful to clear the faults just to see what conditions went away.

11.3.3. RMC70 Command: Initialize Axes (7)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command is used only after the RMC70 is powered up or restarted to initialize the controller. No motion commands are allowed until the axes have been initialized and the controller enabled.

This command does the following:

- Sets the Enabled bit in the Controller Status register.
- Sets the Initialized status bit on all axes.

These steps enable most motion commands and Auto Stops on the RMC70.

This command need not be reissued unless the RMC70 is restarted. This command does not clear any faults.

Note:
The RUN Mode (98) command also initializes the axis, plus puts the controller in RUN mode. See the RUN Mode (98) topic for details.

Command Parameters

None.

Why Bother?

This command keeps the RMC70 from executing commands before you have specifically told it to. Sometimes the RMC may have powered up with invalid data in it. A cleared Initialized bit tells you "Hey, I just woke up! I won't do anything until you initialize me! Then I am convinced that you are aware that I just started up and may be all messed up!"

Checking the Initialized bit is a good way to detect if the RMC70 has been reset from a power outage, for example.

11.3.4. RMC70 Command: Fault Controller (8)

Supported Axes:	All
Firmware Requirement:	1.10 or newer

Description

This command stops all motion on the controller, unless an axis is in the Direct Drive state. It can be used as an emergency stop.

This command does the following:

- **Puts the RMC70 in PROGRAM mode**
  This stops all Tasks and disables the PreScan Table.
- **Halts all axes**
  Performs an Open Loop Halt with Disable Output on each axis.

In order to issue motion commands again, you must put the RMC70 into RUN mode.

To issue this command from RMC70Tools, click the **Fault Controller** ( ) button on the toolbar.

Command Parameters

None.

Why bother?

If you want to shut down the RMC70 because a major problem occurred, use this command, because you can be sure it will stop all motion (unless you are in the Direct

Drive state). If you use a regular Halt instead of this command, the PreScan table can still start a User Program, which may start motion again.

11.3.5. RUN Mode (98)

Supported Axes:	All
Firmware Requirement:	1.10 or newer

Description

This command puts the RMC70 in RUN mode, which starts the PreScan Table and enables running programs on any Task. This command also initializes the axes in the same way as the Initialize Axis (7) command.

In RUN mode, the RMC70 Controller LED is a solid green.

Command Parameters

None.

Issuing the RUN Mode Command

To put the RMC70 in RUN mode, use the RUN Mode (98) command.

To issue this command from RMC70Tools, click the **RUN Mode** button () on the toolbar.

Starting the RMC70 in RUN or PROGRAM mode

By default, the RMC70 starts up in PROGRAM mode. You can set it to start up in PROGRAM or RUN mode:

- In the Project Pane, expand the **RMC70** node.
- Right-click the **Programming** node and click **Properties**.
- Choose the **Startup Mode** you want, then click **OK**.
- Right-click the **Programming** node and click **Download Programs**.

Why Bother?

The RMC70 must be in RUN mode in order to start Tasks and run the PreScan Table. If you issue the Fault Controller command to theRMC70, then it will go into PROGRAM mode, which doesn't allow Tasks or the PreScan Table to run. Use this command to put the RMC70 back into RUN mode.

11.3.6. PROGRAM Mode (99)

Supported Axes:	All
Firmware Requirement:	1.10 or newer

Description

This command puts the RMC70 in PROGRAM mode, which stops all running Tasks and the PreScan Table. Tasks cannot be restarted until the RMC70 is put back into PROGRAM Mode.

In PROGRAM mode, the RMC70 Controller LED slowly flashes green.

Command Parameters

None.

Issuing the PROGRAM Mode Command

To put the RMC70 in PROGRAM mode, use the PROGRAM Mode (99) command.

To issue this command from RMC70Tools, click the **PROGRAM Mode** button (■) on the toolbar.

Starting the RMC70 in RUN or PROGRAM mode

By default, the RMC70 starts up in PROGRAM mode. You can set it to start up in PROGRAM or RUN mode:

- In the Project Pane, expand the **RMC70** node.
- Right-click the **Programming** node and click **Properties**.
- Choose the **Startup Mode** you want, then click **OK**.
- Right-click the **Programming** node and click **Download Programs**.

Why Bother?

Sometimes you might not want Tasks or the PreScan Table to run, such as when you are setting up or programming the RMC70. Use PROGRAM mode then.

11.4. Motion Commands

11.4.1. Motion Commands

The following commands are called *motion commands.* They are grouped by type.

Open Loop

Direct Drive (9)

Open Loop Rate (10)

Open Loop Absolute (11)

Open Loop Relative (12)

Point-to-Point

Move Absolute (20)

Move Relative (21)

Quick Move Absolute (15)

Quick Move Relative (16)

Time Move Absolute (23)

Time Move Relative (24)

Stops

Closed Loop Halt (1)

Open Loop Halt (2)

Disable Drive (3)

Hold Current Position (5)

Stop (6)

Specialty

Speed at Position (36)

Pressure

The Ramp Pressure (41) command is also considered a motion command because it can cause motion.

11.4.2. Open Loop

11.4.2.1. RMC70 Command: Direct Drive (9)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

DANGER: The Direct Drive (9) command disables the safety features on the RMC! Use this command carefully!

Description

This command switches the axis to open loop and ramps the Control Output linearly to the **Requested Output** at the rate specified in the **Ramp Rate**.

Note:
This command is identical to the Open Loop Rate (10) command, except that this command disables all auto stops until another motion command is issued. As such, this command is intended for testing the Control Output and *not* for actual motion control. **For motion control, the Open Loop Rate (10) command should be used instead.**

Command Parameters

1 **Output** (V)

2 **Ramp Rate** (V/sec)

Why Bother?

Use the Direct Drive command to test the Control Output when you are setting up the system.

11.4.2.2. RMC70 Command: Open Loop Rate (10)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command switches the axis to open loop and ramps the Control Output linearly to the **Requested Output** at the rate specified by the **Ramp Rate**.

This command is identical to the Direct Drive (9) command, except that this command does not disable auto stops. Therefore, this command should be used in all cases where open loop is required during the regular machine cycle, and the Direct Drive (9) command should only be used to test the Control Output.

Command Parameters

1 **Requested Output** (V)

2 Ramp Rate (V/sec)

Why Bother?

Use this command when you want to give an Open Loop Control Output to the axis. Open Loop is good for making the axis move when you don't care about going exactly at a certain speed or reaching an exact position.

11.4.2.3. RMC70 Command: Open Loop Absolute (11)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Note:
This command is a specialized Open Loop command. Do not use before fully understanding how it works! To simply issue an Open Loop output, use the Open Loop Rate (10) command.

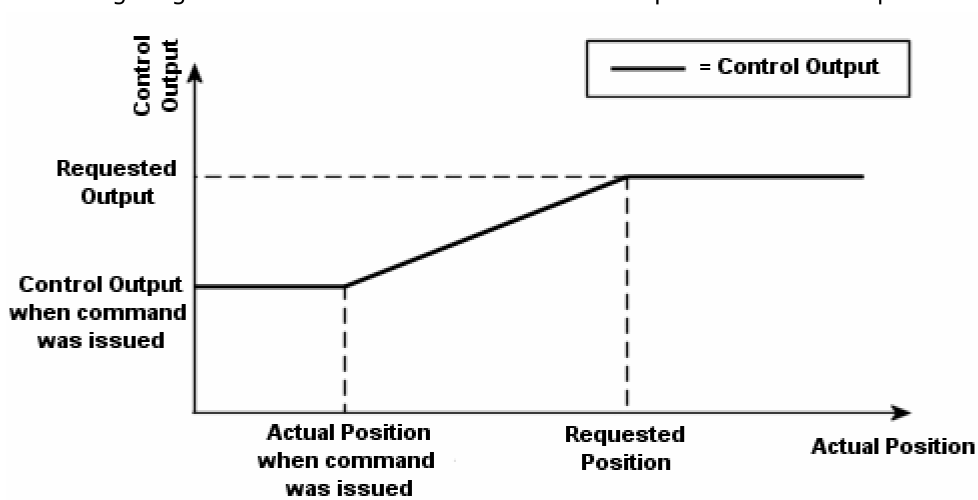
Description

This command first puts the axis in open loop. Then, the RMC70 calculates the Control Output as a function of the Actual Position of the axis. As the Actual Position moves, the Control Output ramps from its value at the time the command was issued to the **Requested Output**. It reaches the **Requested Output** at the **Requested Position**.

Note the following:

- This command does *not* ramp the Control Output based on time.
- If the axis does not move, the Control Output will not change.
- This command cannot be used for starting from a stop because no drive will be applied since the position does not move, and conversely, the position does not move because there is no drive.
- This command is for velocity drive axes only. This command should not be used for a torque drive.

The following diagram illustrates how this command ramps the Control Output:



Command Parameters

1 **Requested Output** (V)

2 **Requested Position** (pos-units)

3 **Direction**

- For linear axes, this parameter must be 0.
- For rotary axes, the value of this parameter defines the direction of the move:
  - o 1 = positive
  - o 0 = nearest *See Special Notes below for more details.*
  - o -1 = negative

Special Notes

Move Direction for Rotary axes.

The behavior of an Open Loop Absolute command on a rotary axis with a **Direction** parameter value of 0 depends on the **Requested Position** of the move and the **Position Unwind** value of the axis.

- If 0 < Requested Position < Position Unwind, then the axis will move in the direction of the closest path to the Requested Position.
- If Requested Position > Position Unwind, then the axis will move in the positive direction.
- If Requested Position < 0, then axis will move in the negative direction.

Why Bother?

This command is often used to stop the axis in Open Loop, instead of an Open Loop Rate. With this command you'll know approximately where the axis will stop, which you don't know with an Open Loop Rate. To do this, the Requested Output must be 0 (or close to zero).

Another use for this command is making an open loop profile based on position, which can be used in injection molding.

Another application is clamping. If the Requested Position is beyond the clamping point, some Control Output will always be applied when it is clamping. When moving into the clamp, the axis will be slowing down because of the ramp and won't hit so hard.

11.4.2.4. RMC70 Command: Open Loop Relative (12)

Supported Axes:	Velocity Drive axes
Firmware Requirement:	1.00 or newer

Note:
This command is a specialized Open Loop command. Do not use before fully understanding how it works! To simply issue an Open Loop output, use the Open Loop Rate (10) command.

Description

This command ramps the open loop Control Output from its current value to the **Requested Output** while the axis moves the distance specified by the **Ramp Distance**.

This command is identical to the Open Loop Absolute (11) except that the Requested Distance is relative to the Actual Position when the command is issued, not an absolute position. See the Open Loop Absolute (11) topic for details.

Command Parameters

1 **Requested Output** (V)

2 Ramp Distance (pos-units, signed)

11.4.3. Point-to-Point

11.4.3.1. RMC70 Command: Move Absolute (20)

Supported Axes:	All position control axes
Firmware Requirement:	1.00 or newer

Description

This command moves the axis in closed loop control to the requested position using the requested speed, acceleration, and deceleration.

Command Parameters

1 **Requested Position** (position-units)

2 **Requested Speed** (position-units/s)

3 **Acceleration Rate** (position-units/s^2)

4 **Deceleration Rate** (position-units/s^2)

5 **Direction**

- For linear axes, this parameter must be 0.
- For rotary axes, the value of this parameter defines the direction of the move:
  - 1 = positive
  - 0 = nearest *See Special Notes below for more details.*
  - -1 = negative

Special Notes

Move Direction for Rotary axes.

The behavior of a Move Absolute command on a rotary axis with a Direction parameter value of 0 depends on the Requested Position of the move and the Position Unwind value of the axis.

- If 0 < Requested Position < Position Unwind, then the axis will move in the direction of the closest path to the Requested Position.
- If Requested Position > Position Unwind, then the axis will move in the positive direction.
- If Requested Position < 0, then axis will move in the negative direction.

Issuing Another Move Command Before Reaching Position

If you issued a Move command and issue another Move command before reaching position and the new Deceleration Rate is smaller than the first one, the target profile may have to overshoot the requested position in order to not exceed the Deceleration rate. In cases where this may be a problem, it is better to use the Speed at Position command with a Requested Velocity of zero. Then the target profile will not overshoot the Requested position.

11.4.3.2. RMC70 Command: Move Relative (21)

Supported Axes:	All position control axes.
Firmware Requirement:	1.00 or newer

Description

This command moves the axis in closed loop control by the **Requested Displacement**, using the **Requested Speed**, **Acceleration Rate**, and **Deceleration Rate**.

This command is similar to the Move Absolute (20) command. However, this command does not use the Direction command parameter, and the Command Position is computed relatively instead of absolutely.

When this command is issued, the **Requested Displacement** is added to the Command Position, if the axis is currently in a mode that uses the Command Position (such as point-to-point moves); otherwise, it is added to the Actual Position.

Command Parameters

1. **Requested Displacement** (pos-units)
2. **Requested Speed** (pos-units/s)
3. **Acceleration Rate** (pos-units/s^2)
4. **Deceleration Rate** (pos-units/s^2)

11.4.3.3. RMC70 Command: Quick Move Absolute (15)

Supported Axes:	All position control axes.
Firmware Requirement:	1.00 or newer

Description

This command ramps the Control Output in open loop from the current value to the specified **Requested Output** at the specified **Open Loop Ramp Rate**. The Control Output stays at that value until the axis must begin decelerating in order to reach the requested position at the specified Deceleration Rate. Once the axis starts decelerating (which can happen while the Control Output is being ramped or while it is constant), the axis switches to closed loop, setting the Target Position and Velocity to the Actual Position and Velocity.

Command Parameters

1. **Requested Position** (pos-units)
2. **Requested Output** (V)
3. **Open Loop Ramp Rate** (V/sec)
4. **Deceleration Rate** (pos-units/s^2)
5. **Direction**
    - For linear axes, this parameter must be 0.
    - For rotary axes, the value of this parameter defines the direction of the move:
        - 1 = positive
        - 0 = nearest *See Special Notes below for more details.*
        - -1 = negative

Special Notes

Move Direction for Rotary axes.

The behavior of a Quick Move Absolute command on a rotary axis with a Direction parameter value of 0 depends on the Requested Position of the move and the Position Unwind value of the axis.

- If 0 < Requested Position < Position Unwind, then the axis will move in the direction of the closest path to the Requested Position.
- If Requested Position > Position Unwind, then the axis will move in the positive direction.
- If Requested Position < 0, then axis will move in the negative direction.

11.4.3.4. RMC70 Command: Quick Move Relative (16)

Supported Axes:	All position control axes.
Firmware Requirement:	1.00 or newer

Description

This command is identical to the Quick Move Absolute (15), except that the Command Position is computed relatively instead of absolutely and that it does not use the Direction command parameter. The **Requested Displacement** is added to the Command Position, if the axis is currently in a mode that uses this value (such as point-to-point moves); otherwise, it is added to the Actual Position.

Command Parameters

1. **Requested Displacement** (pos-units)
2. **Requested Output** (V)
3. **Open Loop Ramp Rate** (V/sec)
4. **Deceleration Rate** (pos-units/s^2)

11.4.3.5. RMC70 Command: Time Move Absolute (23)

Supported Axes:	All position control axes.
Firmware Requirement:	1.00 or newer

Description

This commands moves the axis in closed loop control from the current position to the **Requested Position** in the time specified using a single fifth-order function. Acceleration and velocity are not limited.

Command Parameters

1. **Requested Position** (pos-units)
2. **Time for Move** (ms)
3. **Direction**
   - For linear axes, this parameter must be 0.
   - For rotary axes, the value of this parameter defines the direction of the move:
     - 1 = positive
     - 0 = nearest *See Special Notes below for more details.*
     - -1 = negative

Special Notes

Move Direction for Rotary axes.
The behavior of a Move Absolute command on a rotary axis with a Direction parameter value of 0 depends on the Requested Position of the move and the Position Unwind value of the axis.

- If 0 < Requested Position < Position Unwind, then the axis will move in the direction of the closest path to the Requested Position.
- If Requested Position > Position Unwind, then the axis will move in the positive direction.
- If Requested Position < 0, then axis will move in the negative direction.

11.4.3.6. RMC70 Command: Time Move Relative (24)

Supported Axes:	All position control axes.
Firmware Requirement:	1.00 or newer

Description

This command is identical to the Time Move Absolute (23), except that the command position is computed relatively instead of absolutely and it does not use the Direction command parameter. The **Requested Displacement** is added to the Command Position, if the axis is currently in a mode that uses this value (such as point-to-point moves), otherwise, it is added to the Actual Position.

Command Parameters

1. **Requested Displacement** (pos-units)
2. **Time for Move** (ms)

11.4.4. Specialty

11.4.4.1. RMC70 Command: Speed at Position (36)

Supported Axes:	Position Axes
Firmware Requirement:	1.00 or newer

Description

This command specifies a profile in terms of speed versus position. The RMC70 compares the current Target Position with the **Requested Position** and computes a target profile that reaches the **Requested Velocity** at the **Requested Position**. Once the **Requested Position** (and therefore **Requested Velocity**) is reached, the Done status bit will be set. If the **Requested Velocity** was non-zero, the target generator will continue at the final **Requested Velocity** indefinitely. This command is a closed loop control command.

Command Parameters

1. **Requested Position** (pos-units)
2. **Requested Velocity** (pos-units/sec)

Limitations

1. This command is not intended to be used when the axis is stopped. The axis should be moving at some speed when this command is issued, or it may not work as intended.
2. A **Requested Position** that would require the axis to change direction is not allowed. This will cause a Command Error.

> **Note:**
> A Command Error is cleared when a valid command is issued. If the Auto Stops are not set to halt on a command error, it may be difficult to notice it occurred if a command is issued soon afterwards.

Why Bother?

This command is good when you want to specify the ends, not the means. There are two applications where this really works well:

- When you suddenly want to stop at a certain position, but do not know what deceleration rates you need. Maybe you usually use a deceleration of 100 in/s^2, but if the axis is travelling fast close to the position where you want to stop, the decel rate of 100 might not be enough to stop you fast enough. If you issue a Speed at Velocity (36) command with a Requested Velocity of zero, the RMC70 will calculate the deceleration for you and will get you stopped at the requested position. Note that if you issue this command a long distance before you stop, the calculated decelerations may be so low that it takes a long time to stop.
- Injection molding applications often specify the speed relative to the position. You can set up a user program that issues this command for each trigger point, and it will go to the Requested Velocity at the position.

Status Bits

The Target Generator State Status bits are zero during the Speed At Position command.

11.4.5. Stops

11.4.5.1. RMC70 Command: Closed Loop Halt (1)

Supported Axes:	All axes with a Control Output
Firmware Requirement:	1.00 or newer

Description

This command initiates a Closed Loop Halt. See the Closed Loop Halt topic for details.

Command Parameters

None.

11.4.5.2. RMC70 Command: Open Loop Halt (2)

Supported Axes:	All axes with a Control Output
Firmware Requirement:	1.00 or newer

Description

This command initiates an Open Loop Halt. See the Open Loop Halt topic for details.

Command Parameters

None.

11.4.5.3. RMC70 Command: Disable Output (3)

Supported Axes:	All axes with a Control Output
Firmware Requirement:	1.00 or newer

Description

This command initiates an Open Loop Halt with Disable Output. See the Open Loop Halt with Disable Output topic for details.

Command Parameters

None.

11.4.5.4. RMC70 Command: Hold Current Position (5)

Supported Axes:	All position and pressure control axes.
Firmware Requirement:	1.00 or newer

Description

This command switches the axis to closed loop, unless it already is, and sets the Target and Command Positions equal to the Actual Position. No ramping is done on the Target Position.

Note:
If the axis is moving, this command will stop it suddenly. To slowly stop the axis, use the Stop (6) command.

Command Parameters

None.

11.4.5.5. RMC70 Command: Stop (6)

Supported Axes:	All position, velocity and pressure/force control axes.
Firmware Requirement:	1.00 or newer

Description

This command ramps the current velocity or rate to zero in closed loop. The behavior of this command depends on the type of axis control:

- **Position Control**
  The velocity will ramp down from the current velocity to zero using the specified **Deceleration Rate**, while remaining in position control.
- **Velocity Control**
  The velocity will ramp down from the current velocity to zero using the specified **Deceleration Rate**, while remaining in velocity control.
- **Open Loop**
  The axis transitions to closed-loop control and ramps the velocity to zero using the specified **Deceleration Rate**.
- **Pressure/force Control**
  The pressure/force rate will be ramped to a zero using the specified **Deceleration Rate**.

Command Parameters

1 **Deceleration Rate** (pos-units/s^2)

11.5. System

11.5.1. RMC70 Command: Update Flash (110)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command stores all of the RMC70 controller data to Flash memory for storage in case of power loss. While a FLASH update is in progress, the green CPU LED will flash. Removing power while the LED is still flashing will result in the parameters being lost.

The **Section to Store** parameter must be 0.

Command Parameters

1 **Section to Store** - must be 0

11.6. Pressure Control

11.6.1. RMC70 Command: Enable/Disable Pressure Limit (40)

Supported Axes:	Position/Pressure
Firmware Requirement:	1.00 or newer

Description

This command enables or disables Pressure Limit mode.

In Pressure Limit mode, the Actual Pressure is limited to a certain pressure. This pressure limit is set using the Ramp Pressure (41) command. The axis can be moved normally in Pressure Limit mode as long as the Actual Pressure does not approach the pressure limit. As the Actual Pressure approaches the threshold, the RMC70 will limit the motion so that the pressure threshold is not exceeded.

> **Note:**
> Pressure Limit mode may affect motion even when the Actual Pressure is well below the pressure limit. In order to achieve precise motion when pressure is not important, do not enable Pressure Limit mode. This may require the user to enable Pressure Limit mode only after the pressure has increased close to the point where the pressure is to be limited.

Command Parameters

1 **1 = Enable / 0 = Disable**

Usage Details

> **Note:**
> Controlling pressure is a complex task! See Pressure Control for detailed information on controlling pressure.

Entering Pressure Limit Mode

Use this procedure to properly enter Pressure Limit mode.

1. Move the axis to a point where the pressure is low or zero.
2. Issue a Ramp Pressure (41) command to set the Command Pressure to the desired value.
3. Issue an Enable/Disable Pressure Limit (40) command with a parameter of 1 (one) to enable Pressure Limit Mode.
4. Move the axis as desired. When the motion begins to affect the pressure, the RMC70 will limit the motion such that the Actual Pressure does not exceed the Command Pressure.
5. To change the Command Pressure, issue the Ramp Pressure (41) command.

Controlling Pressure

To make an axis go to a certain pressure, the axis must be commanded to move to a point at or beyond the point where the pressure limit is reached. Use the following procedure:

1. Enter Pressure Limit Mode using the procedure above.
2. Issue a command with a position (or velocity) beyond the point where the Actual Pressure reaches the pressure limit.

Once in Pressure Limit mode, the Ramp Pressure (41) command can be issued to change the pressure limit.

Example

The user wants to move the axis so that it reaches and holds 3000 psi. The user knows that the axis reaches this point at approximately 27 inches. First, the user enters Pressure Limit mode with a pressure limit of 3000 psi. Then, the user issues a Move Absolute (20) command with a Requested Position of 30 inches. The axis moves toward 30 inches but does not get past 27 inches, since the pressure reached 3000 psi at this position.

11.6.2. RMC70 Command: Ramp Pressure(41)

Supported Axes:	Position/Pressure and Pressure
Firmware Requirement:	1.00 or newer

Description

This command ramps the pressure limit to the **Requested Pressure** in the time specified by the **Time for Move** parameter. This command can be used in two ways:

1. To set the pressure limit before entering Pressure Limit mode.
2. To ramp the pressure limit when in Pressure Limit mode.

Command Parameters

1 Requested Pressure (Pr)

2 Time For Move (sec)

Usage Details

See the Usage Details section in the Enable/Disable Pressure Limit (40) command for details.

11.7. Set Parameters

11.7.1. RMC70 Command: Set Integrator Mode (71)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command sets the Integrator Mode. Currently, only two modes are available, Always On and Always Off. This command allows the Integrator mode to be changed at any time.

Command Parameters

1 **0 = off, 1 = on**

11.7.2. RMC70 Command: Set Enable Output (67)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command enables or disables the Enable output.

To specify the behavior of the Enable output switch when the Enable Output is enabled, see the Enable Output Behavior topic.

Command Parameters

1 **1 = Enable / 0 = Disable**

Why Bother?

You can wire the Enable output to turn a motor drive or hydraulic system off. Then, if your Auto Stops are set to Open Loop Halt and Disable Output, which will turn off the Enable output, the drive will shut off when an error occurs. This is a good safety precaution if your transducer fails, for example. Use this command to turn the Enable output back on.

11.7.3. RMC70 Command: Set Actual Position (49)

Supported Axes:	All position feedback axes.
Firmware Requirement:	1.00 or newer

Description

This command sets the Actual Position to the value specified by the **Requested Position** parameter, and adjusts the Target Position by the same amount.

The method of adjusting the Actual Position is axis dependent:

- **Absolute axes:** the Position Offset parameter is adjusted.
- **Incremental axes:** the Actual Position is just an accumulation, so it is simply adjusted directly.

Command Parameters

1 **Requested Position** (pos-units)

11.7.4. RMC70 Command: Set Target Position (48)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command sets the Target Position to the value specified by the **Requested Position** parameter, and adjusts the Actual Position by the same amount.

The method of adjusting the Actual Position is axis dependent:

- **Absolute axes:** the Position Offset parameter is adjusted.
- **Incremental axes:** the Target Position is adjusted directly.

Command Parameters

1 **Requested Position** (pos-units)

11.7.5. RMC70 Command: Read Register (111)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command copies the register pointed to by the **Address** into the axis' Read Response register.

Command Parameters

1 **Address** (integer format - see below)

Register Addresses in Integer Format

The **Address** command parameter must be entered in integer form. See the Register Address Format topic for details on converting form standard address notation to integer.

Why Bother?

This command is used with certain communication types. If the communications is set up to continuously return the Read Response register, then issuing this command will allow you to read whatever register you want. This is a good way of accessing more registers for communication types that are set up to only return certain registers.

11.7.6. RMC70 Command: Write Register (112)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command writes the value specified by the **Value** parameter into the address specified by the **Address** parameter.

Command Parameters

1 **Address** (integer - see below)

2 **Value**

Register Addresses in Integer Format

The **Address** command parameter *must* be entered in integer form. See the Register Address Format topic for details.

11.8. Programming

11.8.1. RMC70 Command: Stop Task (91)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command stops the specified task.

To start a task, see the Start Task (90) command.

Command Parameters

1 **Task Number** - Specifies which task to stop. This value must be an integer from 0 to 3.

What is a Task?

The RMC70 has up to 4 tasks that are numbered 0 to 3. Each Task can run one User Program at a time. The tasks help you keep track of which program is running. See the Tasks topic for more details.

11.8.2. RMC70 Command: Start Task (90)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command starts a Task at the specified User Program. If the specified Task is currently running a User Program, the Task will first be stopped, then it will start immediately at the specified User Program.

Note:
To stop a Task, see the Stop Task (91) command.

Command Parameters

1 **Task Number -** Specifies which task to run the User Program on. The RMC70 has a maximum of 4 tasks. This value must be an integer from 0 to 3.

2 **Program -** Specifies a User Program to start at.

Details

The following items are important when issuing a Start Task command:

- Which axis to issue the Start Task command to

In each step of the User Program you will run, if you have not specified which axis the command will be issued to, then it will be issued to the axis on which the Start Task command was issued. If you *have* specified for each step which axis the command will be issued to, then it does not matter which axis you issue the Start Task command to.

- Selecting which Task to run

  When you issue the Start Task command, you must specify which Task to start. What happens depends on what the Task is doing before you issue the Start Task command:

  - **The Task is stopped.**
    If you select a Task that is stopped, the Task will simply start running at the label you specified.
  - **The Task is running.**
    If you select a Task that is running, it will stop and immediately start at the program you specified.

- Select the User Program

  The second command Parameter, **Program**, specifies which User Program the Task will start at.

Issuing Start Task from a PLC

When issuing the Start Task command from a PLC, the second Command Parameter, Program, requires the Program *number*. The number for each User Program is listed in the Project Pane in the User Programs node.

What is a Task?

The RMC70 has up to 4 tasks that are numbered 0 to 3. Each Task can run one User Program at a time. The tasks help you keep track of which programs are running. See the Tasks topic for more details.

11.9. Plots

11.9.1. RMC70 Command: Enable/Disable Plot Trigger (104)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command enables or disables a plots automatic trigger.

To find out more on triggering, see the Triggering Plots topic.

Command Parameters

1 **Plot Number**

> **Note:**
> A special plot number (-1) is reserved to affect all plots assigned to the axis issuing the command.

2 **0 = disable, 1= enable**

11.9.2. RMC70 Command: Rearm Plot (103)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command re-arms a plot for automatic trigger. It erases any data stored in the plot.

To find out more about re-arming, see the Triggering Plots topic.

Command Parameters

1 **Plot Number**

> **Note:**
> A special plot number (-1) is reserved to affect all plots assigned to the axis issuing the command.

Details

The Re-arm Plot command can be issued from the Plot Manager by clicking the Re-arm Plot  icon in the toolbar.

11.9.3. RMC70 Command: Start Plot (100)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command starts a one-time plot. It erases the plot buffer and starts capturing a plot until the buffer is full. At that point, it stops capturing and waits until the plot is reset. Compare this with the Stop Plot (101) and Trigger Plot (102) commands.

To find out more about plotting, see the Using Plots topic.

Command Parameters

1 **Plot Number**

> **Note:**
> A special plot number (-1) is reserved to affect all plots assigned to the axis issuing the command.

Details

The Start Plot command can be issued from the Plot Manager by clicking the Start Plot  icon in the toolbar.

11.9.4. RMC70 Command: Stop Plot (101)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command stops a continuous plot capture immediately. This allows RMC70Tools to read up the entire plot without any gaps. Plots are either continuously capturing data or done capturing data. Compare this with the Start Plot and Trigger Plot commands.

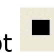
To find out more about plotting, see the Using Plots topic.

Command Parameters

1 Plot Number

> **Note:**
> A special plot number (-1) is reserved to affect all plots assigned to the axis issuing the command.

Details

The Stop Plot command can be issued from the Plot Manager by clicking the Stop Plot  icon in the toolbar.

11.9.5. RMC70 Command: Trigger Plot (102)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command manually triggers a plot. Any post-trigger data is still captured, at which point the data capture stops until the plot is reset. Compare this with the Start Plot and Stop Plot commands.

To find out more on triggering, see the Triggering Plots topic.

Command Parameters

1 Plot Number

> **Note:**
> A special plot number (-1) is reserved to affect all plots assigned to the axis issuing the command.

Details

The Trigger Plot command can be issued from the Plot Manager by clicking the Trigger Plot  icon in the toolbar.

11.10. Step Editor Commands

11.10.1. RMC70 Command: Expression (113)

Supported Axes:	All
Firmware Requirement:	1.00 or newer

Description

This command is used only in the User Program. It evaluates the mathematical **Expression** and assigns the result to a register. Use this command for doing math in the User Program.

Command Parameters

1 Expression

Entering an Expression

Use the Expression Builder to enter an expression.

> **Note:**
> For a full description of building expressions, see the Mathematical Expressions topic.

To open the Expression Builder:

- Choose the Expression (113) command in a step in the User Program.
- Double-click the Expression parameter box.

Building an Expression

An expression consists of registers, operators and functions. Double-click an item to insert it into the expression. To insert a constant value, click **Insert Value.**

An expression *must* begin with a register followed by the ":=" operator. The register must be writable.

Data Types

Expressions follow a strict data type convention. If you mix data types in an expression, you must use the type conversion functions. Note that "3" is a DINT type and "3.0" is a REAL type.

Sample Expressions

This expression	Specifies this action
SampleVariable1 := 2.0 + sqrt(SampleVariable2)	Adds 2 to the square root of SampleVariable2 and assigns the result to SampleVariable1. The variables must exist in the variable table.
_Axis[0].PosOffset:= _Axis[0].ActPos + 2.56	Adds 2.56 to the Actual Position and assigns the result to the Position Offset.
SampleVariable := Min(_Axis[0].ActPos,_Axis[1].ActPos) / 2.0 + 6.0	Divides the minimum value of Axis 0 or Axis 1 Actual Position by 2, adds 6, and assigns the result to the variable SampleVariable. The variable must exist in the variable table.

12.RMC70 Specifications

12.1. RMC70 Specifications

Motion Control	Control loop time	User-selectable 0.5 to 2 ms depending on module configuration.
	Maximum speed	Unlimited
RS-232 Port	Interface with RMC70Tools	Requires a PC with Windows 98/Me/NT/2000/XP.
	Connector	DB-9 Male
	Cable	Null modem
Power	Voltage	+24 VDC +/-10%
	Current – Base only	Typical 290 mA @ 24 VDC, max 375 mA
	Current (4 Expansion modules)	Typical 385 mA @ 24 VDC, max 500 mA
	DC-DC converter isolation	500 VAC, 700 VDC, input to controller
Mechanical	Mounting	Symmetrical DIN 3 or panel-mount
	Dimensions – Base units	3.25 x 5.0 x 2.75 in (8.3 x 12.7 x 7 cm) (WxHxD)
	with 4 Expansion modules	Varies x 5.0 x 2.5 in (?? x 12.7 x 6.4 cm) (WxHxD)
	Weight	12 ounces (0.37 kg) max
	with 4 Expansion modules	Varies, 2.0 lb (0.9 kg) max
Environment	Operating temperature	+32 to +140°F (0 to +60°C)
	Storage temperature	-40 to +185°F (-40 to +85°C)
	Agency compliance	Pending CE, UL, CUL

12.2. Wiring Guidelines

Proper wiring of the RMC70 and of the system is important for proper machine control. Poor wiring is a common source of noisy feedback, drive signals or digital I/O. Follow the guidelines in this topic and the other wiring topics to ensure a low-noise system.

General Wiring Instructions

For CE compliance and to minimize electrical interference:

- Use twisted pairs for all wiring where possible.
- Use shielded cables for all wiring.
- Keep RMC wiring separate from AC mains or conductors carrying high currents, especially high frequency switching power such as conductors between servo drives and motors or amplifiers and proportional valves.

For UL and C-UL compliance:

- Power supply must be Class 2.
- All RMC inputs and outputs must be connected to Class 2 circuits only.

Specific Wiring Instructions

Each RMC CPU Module, Axis Module and Expansion Module has specific wiring diagrams. Follow the links below for wiring details:

CPU Modules

- RMC75S Wiring
- RMC75P Wiring

Axis Modules

- AA Wiring
- MA Wiring

Expansion Modules

- AP Wiring

12.3. RMC70 Part Numbering

Specify RMC70 part numbers when ordering and when contacting Delta customer support.

Part Numbering Schema

(Required for a complete Controller Part Number)		(Optional Expansion modules can be ordered together with the controller or as separate part numbers)			
RMC70 Series CPU	Axis Module	Expansion 1	Expansion 2	Expansion 3	Expansion 4
RMC7NXXX -	**XXN -**	**XXXN -**	**XXXN -**	**XXXN -**	**XXXN**
Choices	**Choices**	**Choices**	**Choices**	**Choices**	**Choices**
RMC75S	MA1	AP2	AP2	AP2	AP2
RMC75P	MA2				

Example Motion Controller Part Numbers

RMC75S-MA2

RMC75P-MA1-AP2

Definitions

Module	Description
CPUs	
RMC75S	Controller with Serial Communications
RMC75P	Controller with Profibus Communications
Axis Modules	
MA1	1-Axis MDT and SSI Inputs
MA2	2-Axis MDT and SSI Inputs
Expansion Modules	
AP2	2- Axis Analog inputs with Pressure Limit

Mounting an RMC70

Mounting Options:

- Symmetrical DIN 3
- Panel-mount

Orientation:

The RMC should be mounted upright on a vertical surface, such that the air holes are on top and bottom.

Clearance above and below:

The amount of clearance required depends on the maximum ambient temperature:

Ambient Temperature	Clearance
122 - 140°F (50-60°C)	3 in (7.6 cm)
86 to 122°F (30 to 50°C)	2 in (5.1 cm)
Less than 86°F (30°C)	1 in (2.5 cm)

13. Index

A

AA Module
about 85
LEDS 88
scaling 89
specifications 85
wiring 86
AA Module 85
Acceleration Feed Forward 138, 171
Actual
acceleration 133
differential force 134
differential force rate 134
position 135
pressure/force 135
force 135
pressure/force rate 136
velocity 136
acceleration cut-off frequency 145
position cut-off frequency 145
pressure/force cut-off frequency 146
pressure/force rate cut-off frequency 146
pressure/force rate 146
velocity cut-off frequency 146
Address
PROFIBUS address 185
register map 114
serial communications address 201
Analog Transducer 85, 102
AP Expansion Module
about 102
LEDs 104
scaling 104
wiring 103
AP Expansion Module 102
Assignment Expressions 42
At Pressure/Force Tolerance 174
At Velocity Tolerance 165
Auto Stops 155
Axes Parameter Editor 68
Axes Status Monitor 63
Axis Modules
AA module 85
about 82
MA module 90
Axis Parameter Registers 145
Axis Parameters pane 68
Axis Status Registers 127
Axis Status Registers pane 63
Axis Tools 60

B

Biasing, RS-485 207

C

Clear Faults Command 217
Closed Loop Control 24
Closed Loop Halt 21
Closed Loop Halt Command 228
Closed Loop Halt Deceleration 157
Command
Position 141
Velocity 141
Pressure/Force 144
about 215
command parameters 215
issue from RMC70Tools 62
issue via serial communications 200
list of commands 215
structure 215
Command 215
Command Tool 62
Commands
Clear Faults (4) 217
Closed Loop Halt (1) 228
Direct Drive (9) 221
Disable Drive (3) 229
Enable/Disable Plot Trigger (104) 235
Enable/Disable Pressure Limit (40) 230
Expression (113) 237
Fault Controller (8) 218
Hold Current Position (5) 229
Initialize Axis (7) 217
Move Absolute (20) 224
Move Relative (21) 224
No-op (0) 217
Open Loop Absolute (11) 222
Open Loop Halt (2) 228
Open Loop Rate (10) 221
Open Loop Relative (12) 223
PROGRAM Mode (99) 219
Quick Move Absolute (15) 225
Quick Move Relative (16) 226
Ramp Pressure (41) 231
Read Register (111) 233
Rearm Plot (103) 236
RUN Mode (98) 219

Set Actual Position (49) 232
Set Enable Output (67) 232
Set Integrator Mode (71) 232
Set Target Position (48) 233
Speed at Position (36) 227
Start Plot (100) 236
Start Task (90) 234
Stop (6) 229
Stop Plot (101) 236
Stop Task (91) 234
Time Move Absolute (23) 226
Time Move Relative (24) 227
Trigger Plot (102) 237
Update Flash (110) 230
Write Register (112) 233
Commands 215
Common Editor Window Components 71
Communications
about 181
PROFIBUS 184
Serial 198
Communications 181
Configuring
Plots 30
Serial Communications 201
Configuring 30
Configuring 201
Connecting RMC70Tools to an RMC70 182
Control axes 2
Control Features 19
Control Modes 23
Control Output 138
Controlling
Force 25
Position-Pressure 26
Pressure 25
Counts 136, 137, 138
CPU Module
about 80
LEDs 80
Custom Data Map Editor 71

D

Damping Factor 174
Data Types 56
Deadband
Deadband Tolerance 158
Output Deadband 160
DF1 Serial Protocol 209
Differential Gain 143, 169
Differential Output Term 139
Differential Term 139
Direct Drive Command 221
Directional Gain Ratio 159
Disable Drive Command 229
Disclaimer xv
Download 62
Drive Output 82

E

Enable Output 83
Enable Output Behavior 163
Enable/Disable Plot Trigger Command 235
Enable/Disable Pressure Limit Command 230
Error
position error 139
velocity error 140
Error Bits 127
Error Codes 107
Errors
error bits 127
error codes 107
Event Log Monitor 61
Expansion Modules
adding 102
Expansion Modules 101
Expressions
about 42
assignment expressions 42
creating 44
functions 44
logical expressions 43
operators 45
troubleshooting 46
Expressions 42
External Halt 21

F

Fault Controller command 218
Fault Input 84
Fault Input Polarity 163
Feed Forwards
Acceleration Feed Forward 171
Acceleration Feed Forward Term 138
Velocity Feed Forward 170
Velocity Feed Forward Term 140
Force A, B Scale 147
Force A,B Offset 147

G

Gains
Acceleration Feed Forward 171
Differential 169
Integral 167
Proportional 167
Velocity Feed Forward 170
Go Online/Offline 68
Graphing 29

H

Halt Group Number 157
Halts

about ... 19
Closed Loop Halt ... 21
External Halt ... 21
Open Loop Halt ... 22
Open Loop Halt with Disable Drive ... 22
Halts ... 19
Help Overview ... xiii
HMIs
communicating with ... 184
HMIs ... 184
Hold Current Position Command ... 229
Host Controller ... 35

I

In Position Tolerance ... 164
Incremental vs. Absolute ... 152
Indirect Data Map ... 181
Indirect Data Map Editor ... 70
Initialize Axes Command ... 217
Inputs
fault input ... 84
Integral Gain ... 167, 175
Integral Output Term ... 139
Integral Term ... 139
Integrator Limit ... 165
Integrator Mode ... 166
Invert Output Polarity ... 160
Issuing Commands ... 215

L

Labels ... 41
Last Error Number ... 129
LEDs ... 80, 96, 104
Line Drivers ... 200
Link Types
about ... 47
conditional jump ... 50
delay ... 48
end ... 48
immediate ... 48
wait for ... 49
Link Types ... 47
Log
Event Log ... 61
Logical Expressions ... 43

M

MA Axis Module
about ... 90
LEDs ... 96
scaling, MDT ... 96
scaling, SSI ... 99
transducer options ... 90
wiring ... 92
MA Axis Module ... 90
Math ... 42
MDT Transducer
fundamentals ... 98
MDT Type ... 149
Menu ... 58
Modbus RTU
about ... 213
register address format ... 114
Modbus RTU ... 213
Move Absolute Command ... 224
Move Relative Command ... 224

N

Natural Frequency ... 173
Negative Pressure/Force Travel Limit ... 179
Negative Travel Limit ... 179
Noise Error Rate ... 152
No-op Command ... 217

O

Offline ... 68
Online ... 68
Open Loop Absolute Command ... 222
Open Loop Control ... 24
Open Loop Halt ... 22
Open Loop Halt Command ... 228
Open Loop Halt Ramp ... 157
Open Loop Halt with Disable Drive ... 22
Open Loop Rate Command ... 221
Open Loop Relative Command ... 223
Output Bias ... 160
Output Deadband ... 160
Output Filter ... 162
Output Limit ... 162
Output Pane ... 63
Output Scale ... 163
Output Velocity ... 139
Outputs
enable output ... 83

P

Parameter Registers ... 145
Part Numbers ... 240
PFID ... 26
PFID Output ... 139
Physical Components Overview ... 1
Plot Manager
about ... 76
screen elements ... 77
Plot Manager ... 76
Plots
about ... 29
advanced plot capabilities ... 29
assigning plots to axes ... 34
configuring ... 30
selecting plot data ... 33
triggering ... 32

viewing 75
Plots...30
Plots...32
Plots...33
Plots...75
Position Differential Gain..............................169
Position Error 139, 164
Position Error Tolerance................................164
Position Integral Gain....................................167
Position Offset...153
Position Proportional Gain 167
Position Scale 153
Position-Pressure Control................................26
Positive Pressure/Force Limit........................179
Positive Travel Limit.....................................178
PreScan Table..55
Pressure Limit..25
Pressure/Force
differential term......................................143
error..143
feed forward term...................................143
proportional term....................................143
rate feed forward term 144
offset...148
scale..148
error tolerance 174
proportional gain 175
integral gain..175
differential gain 176
feed forward 176
rate feed forward 177
PROFIBUS
about 184
configuring..185
mode 16-1 192
mode 8-1 187
mode 8-7 193
PROFIBUS..184
PROGRAM Mode command 219
Program Monitor 64
Programming
overview 51
Programming...51
Project Pane...59
Proportional Gain 167, 175
Proportional Output Term 140

Q

Quick Move Absolute Command...................225
Quick Move Relative Command....................226

R

Raw Counts 137
Raw Counts A, B...138
Read Register Command...............................233
Read Response 130
Rearm Plot Command 236
Reference axes.. 2
Register Address Format
Register Format - Modbus RTU 114
Register Address Format 113
Register Map 114
Registers
status registers 127
parameter registers................................. 145
RMC70
about... 1
axis types.. 2
part numbering 240
RMC70 1
RMC75P CPU Module
about... 79
wiring 79
RMC75P CPU Module 79
RMC75S CPU Module
about... 79
wiring 79
RMC75S CPU Module 79
RMCTools
about... 57
Rotary vs Linear 154
RS-232
Monitor Port.. 182
Wiring 204
RS-232 200
RS-485
Termination... 207
Wiring 205
RS-485 200
RUN Mode command.................................... 219

S

Scaling
about... 11
analog (AA module).................................... 89
analog (AP module) 104
MDT... 96
SSI.. 99
Scaling 11
Serial Communications
about... 198
configuring 201
DF1 protocol 209
issuing commands 200
Modbus Plus protocol 213
network topologies 202
RS-232, RS-485 200
Serial Communications................................. 201
Set Actual Position Command....................... 232
Set Enable Output Command 232
Set Integrator Mode Command 232
Set Target Position Command 233

Shortcut Commands 62
Shortcut Keys 65
Simulate Mode 171
Specifications 239
Speed at Position Command..........................227
SSI
about 100
clock rate..150
format...151
MA module 90
Start Plot Command 236
Start Step Sequence Command......................234
Start-Up Procedure 5
Status Bits...131
Status Registers 127
Step Editor Overview 64
Step labels 41
Stop Command 229
Stop Plot Command.......................................236
Stop Step Sequence Command......................234
Stop Threshold 154
Support 111
System Gain 173

T

Target Acceleration 142
Target Position 142
Target Pressure/Force....................................144
Target Velocity..142
Tasks
about 52
starting 234
stopping...234
Tasks..52
Technical Support..111
Termination
RS-485 207
Time Move Absolute Command 226
Time Move Relative Command 227
Trigger Plot Command 237
Triggering Plots
about 32
Trigger Plot command 237
Triggering Plots...32
Troubleshooting...107
Tuning
about.. 12
tuning a position axis 13
Tuning 12

U

Update Flash Command 230
Upload 62
User Programs
about.. 37
creating.. 38
labels 41
running 40
verifying 40
User Programs 37

V

Variable Table Editor 73
Variables.. 53
Velocity Deadband 154
Velocity Error 140
Velocity Error Tolerance 166
Velocity Feed Forward 170
Velocity Feed Forward Term........................ 140
Velocity Offset 155
Verifying User Programs................................ 40
Viewing
Plots 75
Viewing 75
Volts 137

W

Warranty xv
Wiring
AA module.. 86
guidelines 239
MA module 92
RMC75P 79
RMC75S 79
RS-232 204
RS-485 205
Wiring.. 239
Workspace 57
Write Register Command 233